DP5 Programmer's Guide

Amptek Inc.
14 Deangelo Drive, Bedford, MA 01730
PH: +1 781 275 2242 FAX: +1 781 275 3470
sales@amptek.com www.amptek.com

Note: This document applies to the Amptek DP5, PX5, DP5G, and products derived from these (X123SDD, Gamma-Rad5).

Table of Contents

1 Change List 5
1.1 Programmer's Guide Rev A4 (this document) 5
1.2 FW6.06.00/FP5.11: 5
1.3 FW6.05/FP5.10 5
1.4 FW6.04/FP5.10 5
1.5 FW6.03/FP10 6
1.6 FW6.02/FP5.09 6
2 Installation 6
2.1 Uploading the new firmware FW 6.06 (and FP5.11) 6
2.2 Installing and running the VB Demo 7
3 Interfaces 7
3.1 USB 8
3.1.1 Installing the WinUSB Driver 8
3.2 Ethernet 8
3.2.1 Ethernet Ports 9
3.2.2 General UDP Port 9
3.2.3 Netfinder UDP port 3040 10
3.3 RS232 12
3.4 Transfer times and data buffering deadtime 12
4 Packet Format 14
4.1 Request Packets (Host PC to DP5) 15
4.1.1 Request packet: "Request Status Packet" 17
4.1.2 Spectrum Request Packets 18
4.1.3 Request packet: "Request Digital Scope Data" 19
4.1.4 Request packet: "Request Digital Scope Data & Re-arm Scope" 19
4.1.5 Request packet: "Request Misc Data" 20
4.1.6 Request packet: "Request Ethernet Settings" 21
4.1.7 Request packet: "Request Diagnostic Data" 22
4.1.8 Request packet: "Request Netfinder Packet" 23
4.1.9 Request packet: "I^2C Transfer" 24
4.1.10 Request packet: "Request 32-bit SCA Counters" 24
4.1.11 Request packet: "Latch + Request 32-bit SCA Counters" 25
4.1.12 Request packet: "Latch + Clear + Request 32-bit SCA Counters" 25
4.1.13 Request packet: "Text Configuration" 26
4.1.14 Request packet: "Text Configuration Readback" 27

4.1.15 Request packet: "Clear Spectrum" 28
4.1.16 Request packet: "Enable MCA/MCS" 29
4.1.17 Request packet: "Disable MCA/MCS" 30
4.1.18 Request packet: "Arm Digital Scope" 31
4.1.19 Request packet: "Autoset Input Offset" 32
4.1.20 Request packet: "Autoset Fast Threshold" 33
4.1.21 Request packet: "Write IO3-0" 34
4.1.22 Request packet: "Write 512-byte Misc Data" 35
4.1.23 Request packet: "Set DCAL" 36
4.1.24 Request packet: "Set PZ Correction" 37
4.1.25 Request packet: "Set uC Temperature Calibration" 38
4.1.26 Request packet: "Set ADC Calibration" 39
4.1.27 Request packet: "Clear G.P. Counter" 40
4.1.28 Request packet: "Set Ethernet Settings" 41
4.1.29 Request packet: "Select High-Pass Time Constant" 42
4.1.30 Request packet: "Select RS232 Baud Rate" 43
4.1.31 Request packet: "Set HV Calibration" 44
4.1.32 Request packet: "Set 1.6uS PZ Correction" [PX5 only] 45
4.1.33 Request packet: "Interface Keep-alive – Allow Sharing" 46
4.1.34 Request packet: "Interface Keep-alive – No Sharing" 47
4.1.35 Request packet: "Interface Keep-alive – Lock" 48
4.1.36 Request packet: "Comm test - Request ACK packet" 49
4.1.37 Request packet: "Comm test - Echo packet" 50
4.2 Response Packets 51
4.2.1 Response packet: "Status Packet" 52
4.2.2 Response packet: "256-channel spectrum" 55
4.2.3 Response packet: "256-channel spectrum plus Status" 55
4.2.4 Response packet: "512-channel spectrum" 55
4.2.5 Response packet: "512-channel spectrum plus Status" 55
4.2.6 Response packet: "1024-channel spectrum" 55
4.2.7 Response packet: "1024-channel spectrum plus Status" 55
4.2.8 Response packet: "2048-channel spectrum" 55
4.2.9 Response packet: "2048-channel spectrum plus Status" 55
4.2.10 Response packet: "4096-channel spectrum" 56
4.2.11 Response packet: "4096-channel spectrum plus Status" 56
4.2.12 Response packet: "8192-channel spectrum" 56
4.2.13 Response packet: "8192-channel spectrum plus Status" 56
4.2.14 Response packet: "2048-byte Scope Packet" 57
4.2.15 Response packet: "2048-byte Scope Packet w/ Overflow" 57
4.2.16 Response packet: "512-byte Misc Data Packet" 58
4.2.17 Response packet: "Ethernet Settings" 59
4.2.18 Response packet: "Diagnostic Data" 60
4.2.19 Response packet: "Configuration Readback" 64
4.2.20 Response packet: "Netfinder Packet" 65
4.2.21 Response packet: "I^2C Read Data" 65
4.2.22 Response packet: "64-byte SCA Packet" 66
4.2.23 Response packet: "Comm test - Echo packet" 67
4.3 Acknowledge Packets 68

4.3.1 Acknowledge packet: "OK" 68
4.3.2 Acknowledge packet: "OK, with Interface Sharing Request" 68
4.3.3 Acknowledge packet: "Sync Error" 69
4.3.4 Acknowledge packet: "PID Error" 69
4.3.5 Acknowledge packet: "LEN Error" 69
4.3.6 Acknowledge packet: "Checksum Error" 69
4.3.7 Acknowledge packet: "Bad Parameter" 69
4.3.8 Acknowledge packet: "Unrecognized Command" 70
4.3.9 Acknowledge packet: "PC5 Not Present" 70
4.3.10 Acknowledge packet: "Bad Hex Record" 70
4.3.11 Acknowledge packet: "FPGA Error" 70
4.3.12 Acknowledge packet: "CP2201 Not Found" 70
4.3.13 Acknowledge packet: "Scope Data Not Available" 71
4.3.14 Acknowledge packet: "I^{2}C Error" 71
5 ASCII Commands 72
5.1 Table 4 – ASCII Command Summary 73
5.1.1 AINP - Set the Input Polarity 76
5.1.2 AUO1 - Select AUX_OUT1 Signal 77
5.1.3 AUO2 - Select AUX_OUT2 Signal 78
5.1.4 BLRD - Select the Baseline Restorer 'Down' Correction 79
5.1.5 BLRM - Select the Baseline Restorer Mode 80
5.1.6 BLRU - Select the Baseline Restorer 'Up' Correction 81
5.1.7 BOOT - Set Power-on State 82
5.1.8 CON1 – Select signal for AUX1 Connector 83
5.1.9 CON2 – Select signal for AUX2 Connector 84
5.1.10 CLCK - Select FPGA Clock 85
5.1.11 CUSP - Specify Non-Trapezoidal Shaping 86
5.1.12 DACF - Set DAC Offset 87
5.1.13 DACO - Select Signal for Output DAC 88
5.1.14 GAIA - Set the Analog Gain Index 89
5.1.15 GAIF - Set the Fine Gain 90
5.1.16 GAIN - Set the Total Gain 91
5.1.17 GATE - Configure the GATE Input 92
5.1.18 GPED - Select General Purpose Counter Edge 93
5.1.19 GPGA - General Purpose Counter Uses GATE 94
5.1.20 GPIN - Select the Source for the General Purpose Counter 95
5.1.21 GPMC - General Purpose Counter is Cleared with MCA 96
5.1.22 GPME - General Purpose Counter Uses MCA Enable 97
5.1.23 HVSE - Turn On/Off the PC5 High Voltage Supply 98
5.1.24 INOF - Set the Input Offset 99
5.1.25 INOG - Set the Input Offset Gain 100
5.1.26 MCAC - Select Number of MCA Channels 101
5.1.27 MCAE - Initial State of MCA Enable 102
5.1.28 MCAS - Select the MCA Source 103
5.1.29 MCSL - Set Low Threshold for MCS 104
5.1.30 MCSH - Set High Threshold for MCS 105
5.1.31 MCST - Set the MCS Timebase 106
5.1.32 PAPS - Turn On/Off the Preamp Power Supplies 107

5.1.33 PAPZ – Preamp Pole-Zero Cancellation 108
5.1.34 PDMD - Select the Peak Detect Mode 109
5.1.35 PRCL - Set Low Threshold for Preset Counts 110
5.1.36 PRCH - Set High Threshold for Preset Counts 111
5.1.37 PREC - Preset Counts 112
5.1.38 PRER - Preset Real Time 113
5.1.39 PRET - Preset Acquisition Time 114
5.1.40 PURE - Pile-up Reject Enable 115
5.1.41 RESC - Reset the Configuration to Defaults 116
5.1.42 RESL - Select Reset Lockout Interval 117
5.1.43 RTDD - Set Custom RTD Oneshot Delay 118
5.1.44 RTDE - Enable Risetime Discrimination 119
5.1.45 RTDS - Set Risetime Discrimination Sensitivity 120
5.1.46 RTDT - Set Risetime Discrimination Threshold 121
5.1.47 RTDW - Set Custom RTD Oneshot Width 122
5.1.48 SCAH - Set SCA High Threshold 123
5.1.49 SCAI - Set SCA Index 124
5.1.50 SCAL - Set SCA Low Threshold 125
5.1.51 SCAO - Select SCA Output Level 126
5.1.52 SCAW - Select SCA Output Pulse Width 127
5.1.53 SCOE - Set Digital Scope Trigger Edge 128
5.1.54 SCOG - Set Digital Scope Gain 129
5.1.55 SCOT - Set Digital Scope Trigger Position 130
5.1.56 SOFF - Set Spectrum Offset 131
5.1.57 TECS - Turn On/Off the PC5 Thermoelectric Cooler (TEC) Supply and Set the Temperature 132
5.1.58 TFLA - Select Flat Top Width 133
5.1.59 THFA - Select Threshold for Fast Channel 134
5.1.60 THSL - Select Threshold for Shaped Channel 135
5.1.61 TLLD - Select Threshold for Low-Level Discriminator (LLD) 136
5.1.62 TPEA - Set Peaking Time 137
5.1.63 TPFA - Select Peaking Time for Fast Channel 138
5.1.64 TPMO - Select Test Pulser Mode 139
5.1.65 VOLU – Turn the PX5 speaker ON/OFF 140

1 Change List

Programmer's Guide A4 is released with firmware version FW6.06.00, FPGA version FP5.11, and Visual Basic Demo v2.50, coinciding with the initial release of PX5s.

1.1 Programmer's Guide Rev A4 (this document)

Added transfer times (section 3.4)
New commands and packet types are listed below, and in their corresponding sections

1.2 FW6.06.00/FP5.11:

Channel numbers were changed to start with channel 0, not channel 1, to be consistent with industry standards. This affects the following text commands: MCSL, MCSH, PRCL, PRCH, SCAL, SCAH, TLLD. The data format in the spectrum response packets is unchanged, but software should interpret the first channel to be channel 0, not channel 1, to be consistent with the above commands.
The fine gain now has a minimum precision of 1 part in 8192. (It was as low as 1 part in 256 in FP5.10.) This allows the 'GAIN' and 'GAIF' to be set with the precision of 1 channel or better.
Calibration of the PX5 and DP5/PC5 HV supplies is supported.
PAPZ (preamp pole-zero) command for PX5.
Fixed RTD; in FP5.09/FP5.10, only worked with TPFA (fast peaking time) of 100nS (80MHz) or 400nS (20MHz).
Added a 'build number' to the firmware version, in byte 37 of the status packet; FW6.06, build 0 is listed as FW6.06.00.
Improved precision of THFA (Fast Threshold) and THSL (Shaped Threshold) commands.
Added input offset gain command ('INOG') for PX5

1.3 FW6.05/FP5.10

RS232: A 'Gap timer' was added; support for 57.6kbaud and 19.2kbaud was added, in addition to the standard 115.2kbaud. (See section 3.3).
Added 'reboot' bit to status packet, to aid in detecting unexpected reboot/power cycle.

1.4 FW6.04/FP5.10

UDP receive buffer was expanded to handle 590-byte DHCP response

Fixed THFA readback for unconfigured condition
Fixed 'Unit is Configured' bit in status packet

1.5 FW6.03/FP10

CON1 & CON2 commands for GammaRad (DP5G/PC5G) and PX5

1.6 FW6.02/FP5.09

Shaped channel peaking times ('TPEA') down to 50nS; Fast channel peaking times ('TPFA') down to 50nS

Feedback on this firmware, documentation, and VB demo should be sent to Dave Sperry, dsperry@amptek.com.

2 Installation

2.1 Uploading the new firmware FW 6.06 (and FP5.11)

This only needs to be performed if upgrading a DP5 from FW5.xx to FW6.xx. Once a DP5 has FW6.00 or later, the VB Demo application can be used to upload new firmware via any interface.
Note: WinUSB should be installed (as described in Section 3.1 below) prior to connecting the upgraded DP5's USB cable to the PC.

1. The ZIP file containing the firmware and utilities need to be unzipped on to a local hard drive.
2. Install the DP5Loader application by running SETUP.EXE from \DP5 Upload Utility\SWSETUP
3. Run the DP5Loader application. This requires an RS232 connection to the DP5.
4. Select the COM port the DP5 is connected to. Click the 'Check Device Communications' button. If it locates the DP5, then…
5. Select the Upload Type as FPGA, click 'Upload FPGA/uC HEX File', and select '_dp5_fpga_v511.mcs' from wherever the ZIP file was unzipped
6. When the upload completes, it says the power needs to be cycled on the DP5 – this isn't necessary, but the DP5Loader application has to be closed and reloaded.
7. Once DP5Loader is restarted, click 'Check Device Comm…' again. Then select 'uC' for Upload Type, click 'Upload FPGA/uC HEX File' and select 'DP5_v6.06.00.hex' from wherever the ZIP file was unzipped.
8. Now, when you power-cycle the DP5, it should be running the new firmware (and is no longer compatible with the DP5Loader application or ADMCA).

9. You can leave the RS232 cable connected to verify communications (below), before moving on to USB or Ethernet.

2.2 Installing and running the VB Demo

1. Navigate to wherever the ZIP file was unzipped and locate the 'Setup_v250' folder.
2. Run 'SETUP.EXE' from this folder. This will install the VB5 demo application.
3. Run the application that was installed, 'DP5_v250.exe'.
4. With the RS232 cable connected to the DP5, select the appropriate COM port.
5. Switch to the 'spectrum' tab, and click 'Single Update'. It should acquire data from the DP5 (which is in an unconfigured state.) If the values above the spectrum update, and it displays 'Spectrum OK' in the lower left text box, then it is communicating.
   [Steps 6-8 allow configuring an IP address, if Ethernet will be used.]
6. Go back to the 'Communication' tab, and set the IP Address options on the right. Click 'Send to DP5' when done.
7. Now, you should be able to connect Ethernet to the DP5. The green light on the DP5 Ethernet jack should light and blink to indicate a valid link.
8. Restart the DP5_v250.exe application. (It needs to be restarted to change the communication interface.) Click the 'Search for DP5s' button. It should display a separate form, which after a short time should show the DP5, along with its IP address, etc. Click on it to pass the IP address back to the Communication tab.
9. Click the 'spectrum' tab, and click 'auto update'. It should update the data, indicating that it is able to communicate via Ethernet.
10. Click the 'Configuration' tab. [Currently, the demo app can only send a text configuration file.] Click the 'Send text configuration file' button. For a SDD with 5V preamp, select the 'SDD Config.txt' configuration file from the location where the ZIP file was unzipped. (This file is for an Amptek SDD w/ 5V preamp: HV=-190, TEC=220K). This file can be edited to modify the DP5 parameters (make sure to save the changes before sending it!)

    Note 1: The VB5 source code for the demo is located in the Setup_v250\Source folder.

3 Interfaces

The DP5 supports three communication interfaces: USB, Ethernet and RS232. Details about the specific interfaces are in the following sections.

Note: Attempting to use multiple interfaces simultaneously can result in unexpected behavior. Because of this, the Ethernet general UDP socket is disabled whenever the USB port is connected to a live host port. The socket is re-enabled if the USB cable is disconnected, or the host/hub to which the DP5 is physically connected is powered off. The 'Netfinder' socket remains functional when USB is connected, and the contents of the Netfinder packet indicate if the UDP socket is unavailable because of USB.
The DP5G checks to see if either the USB or RS232 connectors are connected to a live port at power-up. If so, the Ethernet controller is not initialized (to save power), and will remain non-functional until the next power cycle.

3.1 USB

The DP5 supports 'full-speed' (12Mbps) USB 2.0. Three endpoints are used: the 'control' endpoint (EP0); EP1 IN (for response packets from the DP5 to host PC); and EP2 OUT (for request packets from the host PC to DP5.)

Transfers on EP1 and EP2 are terminated by a 'short packet' (a packet smaller than the max packet size of 64 bytes), or a zero-length packet ('ZLP').

For the DP5, PX5, and DP5G, the USB Vendor ID (VID) and Product ID (PID) are:
VID: 0x10C4
PID: 0x842A

The DPPMCA application (and DP5 Visual Basic demo) use the WinUSB driver. This is a Microsoft product that is native to Vista and Windows 7 (32- and 64-bit), and installable on WinXP. The Visual Basic demo includes source code, and VC++ classes are available for those wishing to write their own communication software.

3.1.1 Installing the WinUSB Driver

Currently, the INF file which links the DP5 to the WinUSB driver supports 32-bit WinXP, and 32- and 64-bit versions of Vista and Windows 7. (Vista support has not been tested.) Directions for installing the driver, and the certificate which was used to sign the driver package, are in the 'WinUSB Driver Installation' folder in the ZIP file.

The ZIP file includes the Microsoft document 'WinUsb_HowTo.docx', which describes the complex process of using Windows' SetupAPI to find the device path, etc. [The VB demo demonstrates this technique, as well as setting pipe policies, opening pipes, sending and receiving data, etc.]

USB.h and USB100.h contain many structures, enums, etc. used by Windows' USB implementation – these files (from DDK6001) are in the ZIP file.

Microsoft's website gives information on how to call the specific WinUSB functions:
http://msdn.microsoft.com/en-us/library/ff540046(v=VS.85).aspx

3.2 Ethernet

The DP5 incorporates a 10base-T Ethernet controller and embedded TCP/IP stack to provide Ethernet support via a standard RJ45 jack. The DP5 supports auto-negotiation for half- and full-duplex. Only 10Mbit/s is supported.

PING and ARP are supported. A fixed IP address can be specified, or a dynamic address can be obtained from a DHCP server. (If 'dynamic' is selected, and the DP5 is unsuccessful in obtaining a lease from a DHCP server, it will fall back to its previously-programmed static address.) The IP configuration (fixed/dynamic, IP address, netmask, gateway, and port) can be programmed via any of the DP5 communication interfaces (Ethernet, USB or RS232).

3.2.1 Ethernet Ports

The DP5 supports two UDP ports for communication: one port is for general communications (fixed at UDP port 10001 in FW6.01, and probably programmable in later firmware releases), and the other is for device discovery via the Silicon Labs Netfinder protocol (UDP port 3040.)

3.2.2 General UDP Port

By default, the DP5 uses UDP port 10001 for communications. Once a packet exchange has taken place on this port, the socket is 'bound' to the IP address **and source port** of the host it exchanged packets with. Once the socket is bound, packets from other IP addresses will be ignored. After approx. 4-5 seconds of inactivity on the socket, the socket is reset so that it can once again connect to any IP.

To not allow other hosts to seize control of the DP5, the host can continue exchanging packets so that this timeout does not occur. "Interface keep-alive" Request Packets can be sent periodically, if there is no other periodic traffic, to keep the socket from resetting. Or an "Interface – lock" Request Packet can be sent, in which case the socket will not be reset until power is cycled on the DP5, or the host removes the lock by sending an "Interface keep-alive - allow sharing" or "Interface keep-alive - no sharing" Request Packet.

Broadcast packets can be received by the General UDP Port, which may be useful if multiple DP5s are on a subnet, and synchronization or centralized control is desired. (Request packets such as 'Enable MCA' and 'Clear Spectrum' could be useful as broadcasts.) As of firmware FW6.01, all Request Packets received by the DP5 will generate a Response Packet, which isn't desirable for broadcasts. Future versions of the firmware will use the most significant bit of the PID2 packet ID to indicate a broadcast packet, so that DP5s will be able to suppress Response Packets. Also, a command or Request Packet will be added so that a DP5 can be configured whether to process or ignore broadcast packets. (This will allow some DP5s on a subnet to not participate in broadcasts, if this is desired.)

The existing protocol works over the Internet, but possibly not very well. As it uses raw UDP packets with no sequence numbers, and spectrum packets are large enough that they're split into multiple UDP packets, theoretically the spectrum UDP packets could be received out of sequence, with no way to tell. Eventually, the firmware will have the option to encapsulate each UDP packet within the DP5 packet structure, and use the DP5 packet PIDs as sequence numbers. This should allow the protocol to be more robust when routed over the Internet – the packets will be able to be reordered if they arrive out of sequence, and it will be easier to detect dropped packets.

Timeouts

Generally, the DP5 sends the Response or ACK packet after it has completed the operation specified by the Request Packet. This is usually happens quickly, so that the response is sent with little delay. Since the DP5 always sends a Response or ACK Packet in response to receiving a Request Packet, it may be desirable to use a timeout interval to identify missed packets.

The Visual Basic demonstration software uses a timeout interval of 1000mS by default. However, there are a few packet types which require a longer timeout interval because they may take longer than 1s to execute:

Description	PID1	PID2	LEN MSB	LEN LSB	Recommended Timeout
Request diagnostic data	3	5	0	0	2500mS

Erase FPGA image	0x30	1	0	2	5000mS
Erase uC image #1	0x30	5	0	2	2000mS
Write 512-byte Misc Data	0xF0	0x09	2	0	1000mS

3.2.3 Netfinder UDP port 3040

The DP5 supports the Silicon Labs 'Netfinder' protocol on UDP port 3040. This allows DP5s to be 'discovered' – this is particularly useful if DHCP is used to assign dynamic IP addresses, as the DP5 may have an unknown address.

By broadcasting a Netfinder 'Broadcast Identity Request' to UDP port 3040, the host directs all DP5s that receive the broadcast to respond with an 'Identity Reply', which includes the DP5 serial number, IP address, MAC address, a text description (if one was programmed), the time the device has been powered, and the status of the interface. The host collects the responses from all the DP5s on the subnet, and can then determine which IP address to access based on DP5 serial number, description, etc.

The 'Broadcast Identity Request' is 6 bytes in size and is defined below. The format of the 'Identity Reply' is also listed below – its size is variable because it contains several variable-length null-terminated strings. Note that the 2nd string, the description, comes from the DP5's 'Misc Data' buffer. If the firmware finds a null-terminated string of 40 characters or less, it will send that as the description string. Otherwise, '(no description)' is used. [Note: the DP5 will not respond to the Netfinder 'standard' 4-byte version of the 'Broadcast Identity Request' as defined by Silicon Labs.]

Note that because UDP packet reception is not guaranteed (packets aren't automatically resent), occasionally not all responses will be received. It may be necessary to send multiple Identity Request packets to receive responses to all units. (This depends on how busy the network is, and how many DP5s are present.) Each subsequent Identity Request should use a different sequence ID – a DP5 will not respond to a repeated Identity Request with the same sequence ID.

Note that while the 'Broadcast Identity Request' packet is normally sent as a UDP broadcast packet (i.e. a destination IP of xxx.xxx.xxx.255), it can also be sent to a specific IP address. Also note that UDP broadcast packets generally are limited to the local subnet, and not routed beyond that.

The 'Interface Sharing Request' format is also listed below. If a Netfinder Identity Reply has indicated that a DP5 is in use, but that sharing is allowed, the host can send the 'Interface Sharing Request' to the Netfinder port of that DP5. The DP5 will use an ACK packet to indicate to its current host that a Sharing Request has been received. If that host wishes to grant the request, it does so by stopping transfers to the DP5's UDP socket, which will allow the socket to reset itself and the new host to gain control of it. [No acknowledgement of the 'Interface Sharing Request' is sent by the DP5 – the host can resend it periodically, and either check the Interface Status in the Netfinder 'Identity Reply', or try sending packets to the DP5 UDP port to see if it responds.]

Broadcast Identity Request – 6 bytes

Offset	0	1	2	3	4	5
Value	0x00	0x00	Random 16-bit sequence ID		0xF4	0x FA

Identity Reply – variable length

Data Offset	Value
0	0x01
1	Ethernet General UDP Port Status: 0 = Interface is open (unconnected); 1 = Interface is connected (sharing is allowed); 2 = Interface is connected (sharing is not allowed); 3 = Interface is locked 4 = Interface is unavailable because USB is connected
2-3	Replication of Random 16-bit sequence ID received from 'Identity Request'
4-5	Event 1 Days
6	Event 1 Hours
7	Event 1 Minutes
8-9	Event 2 Days
10	Event 2 Hours
11	Event 2 Minutes
12	Event 1 Seconds
13	Event 2 Seconds
14-19	MAC address (MSB…LSB)
20-23	IP Address (MSB…LSB)
24-27	Subnet Mask (MSB…LSB)
28-31	Default Gateway (MSB…LSB)
variable	Null-terminated string – 'Amptek DP5' + serial number
variable	Null-terminated string – Description/misc text
variable	Null-terminated string – Event 1 description
variable	Null-terminated string – Event 2 description

Interface Sharing Request – 6 bytes

Offset	0	1	2	3	4	5
Value	0x04	0x00	Random 16-bit sequence ID		0xF4	0xFA

3.3 RS232

The RS232 port on the DP5 operates at 115,200 baud, with 1 start bit, 8 data bits, no parity bit, and 1 stop bit. Neither hardware nor software handshaking is used. 57,600 or 19,200 baud operation can be selected; however, selecting these rates will disable the Ethernet port.

The RS232 interface employs a 'gap timer' on its receiver, to aid in synchronization – if the time between consecutive received bytes exceeds 100mS, all buffered bytes are discarded, and the DP5 will resume searching for the sync characters that signal the start of a new packet. No acknowledge packet will be sent to indicate this has occurred.

3.4 Transfer times and data buffering deadtime

In response to a Spectrum (or spectrum + status) request packet, the DP5 will briefly shut off data acquisition, so that it can make a copy of the spectrum memory. When complete, the MCA is re-enabled and the acquisition continues. This 'deadtime' is a function of how many channels are selected (256, 512, etc.) and whether the FPGA is running at 20 or 80MHz. This 'deadtime' will occur each time the spectrum is read out, and this is reflected in the acquisition time; the acquisition timer is stopped during this 'deadtime', while the realtime timer continues to run.

[Note that there is no deadtime penalty for using the ' request spectrum and clear' (or 'request spectrum + status and clear') form of the request packet – the clear function happens simultaneously with the buffering function, and does not add to the 'deadtime'.]

The measured 'deadtimes' are:

# of Channels	20MHz Clock	80MHz Clock
256	228uS	113uS
512	420uS	189uS
1024	804uS	343uS
2048	1.57mS	650uS
4096	3.12mS	1.27mS
8192	6.18mS	2.50mS

The USB, Ethernet and RS232 transfer times were measured, using an oscilloscope. The transfer times include the time from when the request packet was started to when the response packet was completed. (i.e. the whole round-trip transfer was measured.)

The transfer times include the 'deadtime' due to buffering, listed above. The transfer times were measured with the 80MHz clock selected, and were measured using the 'Request spectrum + status' request packet. The transfers will run at the same rate with the 20MHz clock, except the 'deadtime' for buffering will be longer. So, for 20MHz timing, use the 80MHz timing, plus the difference between the 20MHz and 80MHz buffering 'deadtime'.

# of	USB	Ethernet	RS232	RS232

Channels	(80MHz)	(80MHz)	(115k, 80MHz)	(57k, 80MHz)
256	2.8mS	11mS	74mS	146mS
512	3.5mS	19mS	140mS	279mS
1024	4.8mS	35mS	273mS	544mS
2048	7.5mS	70mS	538mS	1.07S
4096	12.7mS	134mS	1.07S	2.14S
8192	24.2mS	263mS	2.14S	4.22S

4 Packet Format

All communications with the DP5 follow a request/response format: the host sends a Request Packet to the DP5, and the DP5 returns a Response Packet. The Response Packet sent by the DP5 to the host will be:

a. a packet containing the requested data;
b. an acknowledge (ACK) packet, indicating the request packet was received and interpreted without error (if the request packet is not one for which data is returned);
c. an ACK packet, indicating the request packet contains an error in structure or content;
d. an ACK packet, indicating why the request can't be executed;
e. an ACK packet, indicating that an ASCII command had a unrecognized command field or an invalid parameter field – the command in question will be echoed in the data field of the ACK packet.
f. an ACK packet, indicating the request packet was received and interpreted without error, and a sharing request was received from another computer via Ethernet;

All packets sent to and from the DP5 use the same basic packet format: a 6-byte header, which defines the type and length of the packet, an optional data field, and a 16-bit checksum. For packets sent to the DP5, the optional data field can be up to 512 bytes, for a maximum packet size of 520 bytes. For packets sent by the DP5 to the host PC, the maximum data field size is 32767 for a maximum packet size of 32775 bytes.

Packet Format

Offset	0	1	2	3	4	5	6...5+LEN	6+LEN	7+LEN
Value	0xF5	0xFA	PID1	PID2	LEN_MSB	LEN_LSB	Data (optional, 0-512 bytes)	CHKSUM MSB	CHKSUM LSB

Fields:

Offset 0 & 1: Sync bytes – these fields have fixed values of 0xF5 and 0xFA.

Offset 2 & 3: Packet ID fields 1 & 2 (PID1 & PID2) – these define the meaning of the packet, as summarized in Table 1, Table 2 and Table 3.

Offset 4 & 5: 16-bit length field (LEN) of the optional data field. If data field is not present, LEN=0.

If LEN =0:

Offset 6 & 7: 16-bit checksum: MSB, then LSB. This is a two's-complement of the 16-bit sum of all bytes prior to the checksum. (i.e. the 16-bit sum of the checksum and all other bytes in the packet is 0.)

If LEN > 0:

Offset 6: The start of the data field, whose length is given by the 16-bit LEN field. The last byte is offset 5+LEN.

Offset 6+LEN, 7+LEN: Checksum, as described above.

4.1 Request Packets (Host PC to DP5)

Table 1

Description	Support*	PID1	PID2	LEN MSB	LEN LSB	Data (optional)	Response
Request status packet	ALL	1	1	0	0		Data packet
Request spectrum	ALL	2	1	0	0		Data packet
Request & clear spectrum	ALL	2	2	0	0		Data packet
Request spectrum + status	ALL	2	3	0	0		Data packet
Request & clear spectrum + status	ALL	2	4	0	0		Data packet
Buffer spectrum		2	5	0	1	1-byte buffer #	ACK packet
Buffer & clear spectrum		2	6	0	1	1-byte buffer #	ACK packet
Request buffer		2	7	0	1	1-byte buffer #	ACK packet
Request digital scope data	ALL	3	1	0	0		Data packet
Request 512-byte misc data	ALL	3	2	0	0		Data packet
Request digital scope data & re-arm scope	ALL	3	3	0	0		Data packet
Request Ethernet settings	ALL	3	4	0	0		Data packet
Request diagnostic data	ALL	3	5	0	0		Data packet
Request hardware description		3	6	0	0		Data packet
Request Netfinder packet	6.01	3	7	0	0		Data packet
Perform I2C transfer	6.01	3	8	varies		I2C transfer description	ACK or Data
Request 32-bit SCA counters	ALL	4	1	0	0		
Latch + Request 32-bit SCA counters	ALL	4	2	0	0		
Latch + Clear + Request 32-bit SCA counters	ALL	4	3	0	0		
Text configuration (to DP5)	ALL	0x20	2	varies		ASCII configuration	ACK packet
Text configuration Readback (from DP5)	ALL	0x20	3	varies		ASCII configuration	Data packet
Erase FPGA image	ALL	0x30	1	0	2	0x12 0x34	ACK packet
Upload packet (FPGA)	ALL	0x30	2	varies			ACK packet
Reinitialize FPGA		0x30	3	0	0		
Erase uC image #0		0x30	4	0	2	0x12 0x34	ACK packet
Erase uC image #1	ALL	0x30	5	0	2	0x12 0x34	ACK packet
Erase uC image #2		0x30	6	0	2	0x12 0x34	ACK packet
Upload packet (uC)	ALL	0x30	7	varies			ACK packet
Switch to uC image #0		0x30	8	0	2	0xA5 0xF1	ACK packet
Switch to uC image #1	ALL	0x30	9	0	2	0xA5 0xF1	ACK packet
Switch to uC image #2		0x30	0x0A	0	2	0xA5 0xF1	ACK packet

Clear Spectrum Buffer	ALL	0xF0	0x01	0	0		ACK packet
Enable MCA/MCS	ALL	0xF0	0x02	0	0		ACK packet
Disable MCA/MCS	ALL	0xF0	0x03	0	0		ACK packet
Arm digital oscilloscope	ALL	0xF0	0x04	0	0		ACK packet
Autoset input offset	ALL; D, P	0xF0	0x05	0	0		ACK packet
Autoset fast threshold	ALL	0xF0	0x06	0	0		ACK packet
Read IO3-0		0xF0	0x07	0	0		
Write IO3-0	ALL	0xF0	0x08	0	1		ACK packet
Write 512-byte Misc Data	ALL	0xF0	0x09	2	0		ACK packet
Set DCAL	ALL; D, P	0xF0	0x0A	0	2	LSB, MSB	ACK packet
Set PZ correction	ALL	0xF0	0x0B	0	1	PZ correction	ACK packet
Set uC temp cal	ALL	0xF0	0x0C	0	1	uC temp offset	ACK packet
Set Boot flags		0xF0	0x0D	0	2?		ACK packet
Set ADC Cal (gain/offset)	ALL	0xF0	0x0E	0	2	gain, offset	ACK packet
Clear G.P. Counter	ALL	0xF0	0x10	0	0		ACK packet
Set Ethernet settings	ALL	0xF0	0x11	0	0x13	Fixed/Dyn, IP[4], Mask[4], Gateway[4], Dest. IP[4], Port[2]	ACK packet
Select high-pass time constant	6.01	0xF0	0x12	0	1	Time constant	ACK packet
Select RS232 baud rate	6.05	0xF0	0x13	0	1	Baud rate	ACK packet
Set HV Cal (gain/offset)	6.05; D, P	0xF0	0x14	0	2	gain, offset	ACK packet
Interface keep-alive - allow sharing	ALL	0xF0	0x20	0	0		ACK packet
Interface keep-alive - no sharing	ALL	0xF0	0x21	0	0		ACK packet
Interface keep-alive - lock	ALL	0xF0	0x22	0	0		ACK packet
Latch SCAs							
Latch + clear SCAs							
Clear SCAs							
Comm test - Request ACK packet	ALL	0xF1	0…15	0	0		ACK packet
Comm test - Echo packet	ALL	0xF1	0x7F	varies		Data to be echoed	Data packet

Items in italics are considered calibration values and don't normally need to be changed

* Initial firmware release which supports this packet type;
If not support by all three DPP types, then:
D = supported by DP5,
P = supported by PX5, and
G = supported by DP5G

4.1.1 Request packet: "Request Status Packet"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	**0xF5**	**0xFA**	**1**	**1**	**0**	**0**	**0xFE**	**0x0F**

Purpose: This request directs the DP5 to construct a status packet and send it to the host PC. The status packet can be requested with this request packet, or with one of the 'Request Spectrum plus Status' variants, in which case the status packet is included in the packet data field along with the spectrum data.

The status packet contains information about the current state of the spectrum, and of the DP5. Its format is listed with the "Status Packet" Response Packet.

Response: If no errors are detected, a "Status Packet" Response Packet will be returned.

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of the requested data.

4.1.2 Spectrum Request Packets

Request packet: "Request Spectrum"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	**0xF5**	**0xFA**	**2**	**1**	**0**	**0**	**0xFE**	**0x0E**

Request packet: "Request and clear Spectrum"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	**0xF5**	**0xFA**	**2**	**2**	**0**	**0**	**0xFE**	**0x0D**

Request packet: "Request Spectrum plus Status"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	**0xF5**	**0xFA**	**2**	**3**	**0**	**0**	**0xFE**	**0x0C**

Request packet: "Request and clear Spectrum plus Status"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	**0xF5**	**0xFA**	**2**	**4**	**0**	**0**	**0xFE**	**0x0B**

Purpose: These requests direct the DP5 to take a 'snapshot' of the spectrum and corresponding status data and begin transmitting that data. If the 'Request and Clear' form of this packet is send, it will also clear the spectrum and associated status data, and if the MCA was enabled when this request was received, then a new acquisition will start. [Where should the details about what status fields get cleared be discussed?]

Response: If no errors are detected, than either a 'spectrum' response packet, or a 'spectrum plus status' response packet will be sent by the DP5.

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of the requested data. If the FPGA failed initialization, then an FPGA error ACK packet will be returned.

4.1.3 Request packet: "Request Digital Scope Data"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	**0xF5**	**0xFA**	**3**	**1**	**0**	**0**	**0xFE**	**0x0D**

4.1.4 Request packet: "Request Digital Scope Data & Re-arm Scope"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	**0xF5**	**0xFA**	**3**	**3**	**0**	**0**	**0xFE**	**0x0B**

Purpose: These request that the DP5 send a 2048-byte digital scope packet. The second form of the command will also re-arm the digital scope.

Response: If no errors are detected, the DP5 will respond with a '2048-byte Scope Packet' Response Packet (PID1=0x82, PID2=1) or a '2048-byte Scope Packet w/ Overflow' Response Packet (PID1=0x82, PID2=3). (See the Response Packet descriptions for details.)

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of the requested data. Also, if no data is available because the scope hasn't triggered, a "Scope data not available" ACK packet will be returned instead of the requested data.

4.1.5 Request packet: "Request Misc Data"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	**0xF5**	**0xFA**	**3**	**2**	**0**	**0**	**0xFE**	**0x0C**

Purpose: This requests that the DP5 send the 512-bytes of 'miscellaneous data' that is stored in the DP5's non-volatile memory. (See also the "Write Misc Data" Request Packet for information on to write this data to the DP5.)

Response: If no errors are detected, the DP5 will respond with the "512-byte Misc Data" Response Packet (PID1=0x82, PID2=2).

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of the requested data.

4.1.6 Request packet: "Request Ethernet Settings"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	**0xF5**	**0xFA**	**3**	**4**	**0**	**0**	**0xFE**	**0x0A**

Purpose: This requests that the DP5 send the Ethernet settings that are stored in the DP5's non-volatile memory. (See also the "Set Ethernet Settings" Request Packet for information on how to write these settings.)

Response: If no errors are detected, the DP5 will respond with the "Ethernet Settings" Response Packet (PID1=0x82, PID2=4).

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of the requested data. Additionally, if no Ethernet controller is detected on the DP5, the "CP2201 not found" ACK packet will be returned instead of the requested data.

4.1.7 Request packet: "Request Diagnostic Data"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	**0xF5**	**0xFA**	**3**	**5**	**0**	**0**	**0xFE**	**0x09**

Purpose: This requests that the DP5 perform some diagnostic tests, and return a suite of diagnostic data.

Response: If no errors are detected, the DP5 will respond with the "Diagnostic Data" Response Packet (PID1=0x82, PID2=5). Note: the DP5 runs a full memory test on the 512KB SRAM, which takes some time. The DP5 will respond with the Response Packet a maximum of 2.5s after receiving the Request Packet.

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of the requested data.

4.1.8 Request packet: "Request Netfinder Packet"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	**0xF5**	**0xFA**	**3**	**7**	**0**	**0**	**0xFE**	**0x07**

Purpose: This requests a Netfinder packet, but through the active communications port. [The 'Netfinder packet' is generally obtained via a broadcast on the TCP/IP Netfinder socket. See section 3.2.3 for details.]

Response: If no errors are detected, the DP5 will respond with the "Netfinder Packet" Response Packet (PID1=0x82, PID2=8).

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of the requested data.

4.1.9 Request packet: "I²C Transfer"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6…5+LEN	6+LEN	7+LEN
Value	**0xF5**	**0xFA**	**3**	**8**	**varies**		**varies**	**varies**	

Purpose: The data field of this Request Packet contains the description for an I²C transfer to be performed by the DP5. Read, Write and Combined I²C transfers are supported.

The format of the data field is listed below. The first 3 bytes are mandatory; following those are the bytes to be written, which can number from 0 to a maximum of 32. Therefore, the size of the data field (LEN) has a minimum of 3 and a maximum of 35.

Data Offset	Allowed Values	Value
0	1-127	7-bit I²C slave address ('SA')
1	0-32	Number of bytes to read ('NOBR')
2	0-32	Number of bytes to write ('NOBW')
3-34	0-255	Bytes to be written (optional)

Note: The DP5 uses Slave Addresses of 0x20 and 0x4C. The optional PC5 uses Slave Addresses of 0x1F, 0x34, 0x50 and 0x6D. Transfers should not be attempted using these Slave Addresses. Also, the value in offset 0 is the 7-bit address – this value will be shifted left one bit, and the appropriate R/W bit merged in, before it's transmitted on the I²C bus.

Response: If no errors are detected, the DP5 will respond either with an ACK OK packet (if NOBR=0), or with an I²C Response Packet with the bytes read from the I²C slave (if NOBR > 0).

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of the requested data or the ACK OK packet. If the I²C master didn't detect the I²C ACK at the proper time, an 'I²C ERROR' ACK packet will be returned. A BAD PARAMETER ACK packet will be returned if there are structure problems with the request. (NOBR=NOBW=0; LEN not equal to NOBW+3; NOBR > 32; NOBW > 32; or SA = 0).

4.1.10 Request packet: "Request 32-bit SCA Counters"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	**0xF5**	**0xFA**	**4**	**1**	**0**	**0**	**0xFE**	**0x0C**

4.1.11 Request packet: "Latch + Request 32-bit SCA Counters"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	**0xF5**	**0xFA**	**4**	**2**	**0**	**0**	**0xFE**	**0x0B**

4.1.12 Request packet: "Latch + Clear + Request 32-bit SCA Counters"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	**0xF5**	**0xFA**	**4**	**3**	**0**	**0**	**0xFE**	**0x0A**

Purpose: These request that the DP5 send 64 bytes of SCA counter data (16 SCAs x 32 bits each). The data is read from the SCA counter latches, so the SCAs must have been latched, either via the 'Latch' form of these requests, a 'Latch SCA' Request Packet (which is TBD in FW6.01), or a hardware signal (also TBD.) The 'Clear' form of these requests will clear the SCA counters after the data has been latched.

Response: If no errors are detected, the DP5 will respond with a "64-byte SCA Packet" Response Packet (PID1=0x83, PID2=1).

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of the requested data.

4.1.13 Request packet: "Text Configuration"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6…5+LEN	6+LEN	7+LEN
Value	**0xF5**	**0xFA**	**0x20**	**2**	**varies**		**varies**	**varies**	

Purpose: This sends a Text Configuration packet to the DP5, which the DP5 interprets and executes. (See the section on ASCII Commands for information on individual commands.) The commands are packed into the Data field, with the following rules:

1. Alphabetic characters must be uppercase.
2. Each command consists of a 4-character command, followed by "=", followed by the parameter (max of 10 characters), and terminated by a semicolon (Ex: "TPEA=10;")
3. Units may be included with the parameter, but they are ignored, and count towards the 10-character maximum length for the parameter. (Ex: "TPEA=10US;")
4. The commands are packed together with no whitespace of any kind (tabs, spaces, CR/LF, etc.)
5. **For some commands, the ordering is important**. See the 'ASCII Commands' section for more discussion of this.
6. If all commands fit in the max 512 byte data field, then the configuration can be sent as a single packet. If they don't fit, then additional Text Configuration packets can be sent. Generally, the ASCII Command "RESC=Y;" would be the first command of the first packet, and it would not appear in later packets. If additional packets are needed, ensure that commands are not chopped between packets – only complete command/parameter strings are allowed.

   Here's an example of the data field:
   RESC=YES;CLCK=AUTO;TPEA=10US;TFLA=0US;CUSP=50%;
   RESL=3000US;PDMD=NORM;THSL=1.00%;THFA=40;

Response: If no errors are detected, the DP5 will respond with an ACK OK packet. This indicates that all commands were recognized and executed without error.

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of the requested data. If one or more errors are detected while parsing the Text Configuration, an 'Unknown/Bad Parameter', 'Unrecognized Command', or 'PC5 Not Present' ACK packet will be returned, and the data field will contain the ASCII command and parameter which caused the error. If more than one error was detected, only the last error & command will be returned.

4.1.14 Request packet: "Text Configuration Readback"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6…5+LEN	6+LEN	7+LEN
Value	**0xF5**	**0xFA**	**0x20**	**3**	**varies**		**varies**	**varies**	

Purpose: This sends a Text Configuration Readback Request Packet to the DP5, which the DP5 uses as a 'template' to read back the current configuration. The data field is the same format as for the Text Configuration Request Packet – a list of ASCII command and parameters, separated by semicolons. The DP5 strips off any parameters and replaces them with the current settings for each command specified. The result is returned in the Configuration Readback Response Packet. It is possible to send a text configuration packet via the Text Configuration Request Packet (PID1=0x30 / PID2=2), and then reuse the same packet data field for the Text Configuration Readback Request Packet (PID1=0x30 / PID2=3) to read back the actual settings that the DP5 derived from the Text Configuration packet.

The "=" and parameter are optional for each command listed; the minimum required is the 4-character commands, separated by semicolons. There is one exception to this – the SCAI (SCA Index) command. As this is a directive to the command processor rather than a true command, the '=' and parameter are required. Here is an example data field, to read back the low and high thresholds for SCA1 & SCA2: "SCAI=1;SCAL;SCAH;SCAI=2;SCAL;SCAH;"

Also note that the 'RESC' (Reset Configuration) has no meaning in the context of Configuration Readback – if included, it will return 'RESC=?;'.

Any number of commands may be read back, as long as they fit within the maximum data field of 512 bytes. As of FW6.01, there is no error handling for the condition where the generated Response Packet exceeds the maximum data field size, so limiting the size of the Request Packet data field is a good idea, to avoid this issue.

Response: If no errors are detected, the DP5 will respond with a Configuration Readback Response Packet (PID1=0x82 / PID2 = 7). Any command which is not recognized in the data field will return "CMDA=??;", where 'CMDA' is the unknown command. [This is returned in the Response Packet with the rest of the commands.]

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of the requested data. If one or more errors are detected while parsing the Text Configuration, an 'Unknown/Bad Parameter', 'Unrecognized Command', or 'PC5 Not Present' ACK packet will be returned, and the data field will contain the ASCII command and parameter which caused the error. If more than one error was detected, only the last error & command will be returned.

4.1.15 Request packet: "Clear Spectrum"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	**0xF5**	**0xFA**	**0xF0**	**1**	**0**	**0**	**0xFD**	**0x20**

Purpose: This commands the DP5 to clear the MCA (spectrum), and all associated values (slow counts, fast counts, acquisition time, etc.) If the acquisition was enabled when this command was received, it will continue to be enabled after the spectrum is cleared.

Response: If no errors are detected, the DP5 will respond with ACK OK packet.

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of ACK OK.

4.1.16 Request packet: "Enable MCA/MCS"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	**0xF5**	**0xFA**	**0xF0**	**2**	**0**	**0**	**0xFD**	**0x1F**

Purpose: This commands the DP5 to enable the MCA. If the acquisition was stopped due to a "Disable MCA/MCS" command or because the 'Preset Time' expired, this command will re-enable the acquisition, without clearing anything. If the acquisition was stopped due to 'Preset Counts' being reached, then this command will have no effect until the spectrum is cleared. (See ASCII Commands PREC & PRET.)

Response: If no errors are detected, the DP5 will respond with ACK OK packet.

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of ACK OK.

4.1.17 Request packet: "Disable MCA/MCS"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	**0xF5**	**0xFA**	**0xF0**	**3**	**0**	**0**	**0xFD**	**0x1E**

Purpose: This commands the DP5 to disable the MCA, effectively pausing the acquisition.

Response: If no errors are detected, the DP5 will respond with ACK OK packet.

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of ACK OK.

4.1.18 Request packet: "Arm Digital Scope"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	**0xF5**	**0xFA**	**0xF0**	**4**	**0**	**0**	**0xFD**	**0x1D**

Purpose: This commands the DP5 to arm its internal digital oscilloscope. Once armed, a trigger condition will cause the digital scope to capture 2048 samples. After arming the digital scope, the application software can either:

1. Monitor the 'Oscilloscope data ready' bit (offset 35: bit D2 in the status data) to determine when the scope has triggered, then send a "Request Digital Scope Data" request packet to get the data.
2. Repeatedly send a "Request Digital Scope Data" request packet – the DP5 will respond with a "Scope data not available" ACK packet if the trigger hasn't occurred yet. It will respond with the "2048-byte scope packet" or "2048-byte scope packet w/ overflow" after the scope has triggered.

   (See the DACO, SCOE, SCOT & SCOG ASCII commands for more information on the Digital Scope.)

Response: If no errors are detected, the DP5 will respond with ACK OK packet.

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of ACK OK.

4.1.19 Request packet: "Autoset Input Offset"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	**0xF5**	**0xFA**	**0xF0**	**5**	**0**	**0**	**0xFD**	**0x1C**

Purpose: This commands the DP5 to search for an appropriate input offset (i.e. the DC offset of the input signal as seen by the input ADC.) Generally, the 'default' input offset should work with most detectors. (See the INOF ASCII command.) In order for the Autoset Input Offset function to work properly:

1. The input polarity must be set properly (see the AINP ASCII command.)
2. HV & detector temperature (if applicable) must be reasonably stable.
3. An appropriate input offset might not be found if this command is initiated while the input count rate is high.

Support: This request is supported by the DP5 and PX5, but not the DP5G. (The DP5G does not have a variable input offset.)

Response: If no errors are detected, the DP5 will respond with ACK OK packet. The DP5 sends the ACK packet immediately, and does not wait for the Autoset function to complete. The result of this command is in the status data (offset 36: bit D7) – it returns "Auto Input Offset locked" or "Auto Input Offset searching". Status can be polled until the result indicates 'locked'.

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of ACK OK.

4.1.20 Request packet: "Autoset Fast Threshold"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	**0xF5**	**0xFA**	**0xF0**	**6**	**0**	**0**	**0xFD**	**0x1B**

Purpose: This commands the DP5 to search for an appropriate fast threshold setting. In order for the Autoset Fast Threshold function to work properly, no input counts can be present (other than noise.)

Response: If no errors are detected, the DP5 will respond with ACK OK packet. The DP5 sends the ACK packet immediately, and does not wait for the Autoset function to complete. The result of this command is in the status data (offset 35: bit D6) – it returns "Auto Fast Threshold not locked" or "Auto Fast Threshold locked". Status can be polled until the result indicates 'locked'. The fast threshold determined by this function is returned by the readback of the 'THFA' command, using the 'Text Configuration Readback' Request Packet. Note that this setting is overwritten when a new configuration packet is received with the "THFA" or "RESC" commands.

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of ACK OK.

4.1.21 Request packet: "Write IO3-0"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7	8
Value	**0xF5**	**0xFA**	**0xF0**	**8**	**0**	**1**	**IO3…IO0**	**varies**	

Purpose: This commands the DP5 to set the four IO lines IO3-0 with the data byte in offset 6, as shown below. Note that these outputs are open-drain, with a weak pullup. They can sink 20mA, but only source 100uA (typ) at 3.3V. (These are outputs from a Maxim MAX7328 operating at 3.3V – see the datasheet for more info.)

Bit	Signal	Connector Location
D7-D4	-	N/A
D3	IO3	J6.14 (Auxiliary connector)
D2	IO2	J6.13 (Auxiliary connector)
D1	IO1	J5.18 (Interconnect connector)
D0	IO0	J5.17 (Interconnect connector)

Response: If no errors are detected, the DP5 will respond with ACK OK packet.

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of ACK OK.

4.1.22 Request packet: "Write 512-byte Misc Data"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6-517	518	519
Value	**0xF5**	**0xFA**	**0xF0**	**9**	**2**	**0**	**Misc Data**	**varies**	

Purpose: This commands the DP5 to store the 512-byte data field in non-volatile memory. The data can be binary, ASCII, etc. (It could contain the description or location of the unit as an ASCII string, for example.) It is read back with the "Request Misc Data" Request Packet.

Response: If no errors are detected, the DP5 will respond with ACK OK packet. Note this command requires a FLASH memory erase operation – the ACK packet could be delayed as much as 400mS from the request packet.

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of ACK OK.

4.1.23 Request packet: "Set DCAL"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6-7	8	9
Value	**0xF5**	**0xFA**	**0xF0**	**0x0A**	**0**	**2**	**DCAL LSB, MSB**	**varies**	

Purpose: This commands the DP5 to set the non-volatile calibration value for the detector temperature measurement diode. The setting is an unsigned integer in the range of 0-4095, in the format LSB followed by MSB. The nominal value is 1.000V, which corresponds to a setting of 0x555. (The conversion factor is 732uV/count.) This value can be read back via the "Request Diagnostic Packet" Request Packet. The setting is stored in the DP5 non-volatile memory. If a PC5 is present, it is also stored in the PC5 non-volatile memory.

Support: This request is supported by the DP5 and PX5, but not the DP5G.

Response: If no errors are detected, the DP5 will respond with ACK OK packet.

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of ACK OK.

4.1.24 Request packet: "Set PZ Correction"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7	8
Value	**0xF5**	**0xFA**	**0xF0**	**0x0B**	**0**	**1**	**PZ Corr**	**varies**	

Purpose: This commands the DP5 to set the non-volatile correction factor for the Pole-Zero. This allows digital fine-tuning of the analog PZ time constant. (This calibration value is set at the factory and shouldn't need to be changed.) This value can be read back via the "Request Diagnostic Packet" Request Packet. The interpretation of the PZ Correction value ('setting') is:

Setting of 0-127: the correction factor is (1024+setting)/1024 (i.e. 100% to 112.4%)
Setting of 128-255: the correction factor is (768+setting)/1024 (i.e. 87.5% to 99.9%)
Examples: setting = 0 -> 100% correction factor;
setting =10 -> 101% correction factor;
setting =250-> 99.4% correction factor

The PX5 has two software-selectable pole-zero filters. This command sets the calibration for the 3.2uS pole-zero. The DP5 and DP5G have a single pole-zero; this command sets the calibration for it.

Note: This command won't take effect until the next configuration request packet is received.

Response: If no errors are detected, the DP5 will respond with ACK OK packet.

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of ACK OK.

4.1.25 Request packet: "Set uC Temperature Calibration"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7	8
Value	**0xF5**	**0xFA**	**0xF0**	**0x0C**	**0**	**1**	**uC Temp Cal**	**varies**	

Purpose: This commands the DP5 to set the non-volatile offset calibration for the DP5 temperature measurement. It is an 8-bit signed value, and is added to the measured temperature; the result appears in the status data as 'board temperature' (offset 34). This value can be read back via the "Request Diagnostic Packet" Request Packet. [The temperature sensor is internal to the microcontroller; this calibration value allows compensation for the fact that the uC die is always hotter than the board.]

Response: If no errors are detected, the DP5 will respond with ACK OK packet.

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of ACK OK.

4.1.26 Request packet: "Set ADC Calibration"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6-7	8	9
Value	**0xF5**	**0xFA**	**0xF0**	**0x0E**	**0**	**2**	**Gain, Offset**	**varies**	

Purpose: This commands the DP5 to set the non-volatile gain and offset calibration values for the DP5 ADC. These values can be read back via the "Request Diagnostic Packet" Request Packet.

Gain: If gain setting is 0-127: Correction = (1024+setting)/1024 (i.e. 100% to 112.4%)
If gain setting is 128-255: Correction = (768+setting)/1024 (i.e. 87.5% to 99.9%)
Offset: The offset setting is a signed byte.
Calibrated ADC Value = (Raw ADC Value * Gain) + Offset.

Note: This command won't take effect until the next configuration request packet is received.

Response: If no errors are detected, the DP5 will respond with ACK OK packet.

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of ACK OK.

4.1.27 Request packet: "Clear G.P. Counter"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	**0xF5**	**0xFA**	**0xF0**	**0x10**	**0**	**0**	**0xFD**	**0x11**

Purpose: This commands the DP5 to clear the General Purpose Counter ("G.P. Counter"). (See the ASCII commands GPIN, GPED, GPME, GPGA and GPMC for more information on the G.P. Counter.)

Response: If no errors are detected, the DP5 will respond with ACK OK packet.

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of ACK OK.

4.1.28 Request packet: "Set Ethernet Settings"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6-24	25	26
Value	**0xF5**	**0xFA**	**0xF0**	**0x11**	**0**	**0x13**	**Ethernet settings**	**varies**	

Purpose: This commands the DP5 to set the IP address, etc. for the Ethernet controller. The current values can be read back via the "Request Ethernet Settings" Request Packet.

The format of the data field is:

Data Offset	Value
0	0 = Use fixed IP address; 0xFF = Get IP address from DHCP server
1-4	IP Address (MSB…LSB)
5-8	IP Subnet Mask (MSB…LSB)
9-12	IP Gateway Address (MSB…LSB)
13-16	Reserved
17-18	Port (MSB,LSB: fixed at 10001 for FW6.01)

Response: If no errors are detected, the DP5 will respond with ACK OK packet.

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of ACK OK. Additionally, if no Ethernet controller is detected on the DP5, the "CP2201 not found" ACK packet will be returned instead of ACK OK.

4.1.29 Request packet: "Select High-Pass Time Constant"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7	8
Value	**0xF5**	**0xFA**	**0xF0**	**0x12**	**0**	**1**	**Time Constant**	**varies**	

Purpose: This selects which high-pass time constant the DP5's front end is set for. (The time constant is determined by what resistor values are installed – this Request Packet is used to configure the firmware appropriately. The setting is saved in non-volatile memory, so this Request Packet only needs to be sent if the time constant is changed.) This value can be read back via the "Request Diagnostic Packet" Request Packet. The interpretation of the high-pass time constant value ('setting') is:

0 => 1.625uS time-constant;
1 => 3.225uS time-constant;
2 => 6.425uS time-constant;
3 => 12.825uS time-constant;
4 => 25.625uS time-constant

The PX5 has two selectable poles – 1.625uS and 3.225uS. This command will switch the appropriate pole into the front end. If >3.225uS is selected, then the PX5 will select the 3.2uS slot – component changes are required to convert that slot to the selected time constant. Contact Amptek for more information.

Note: This command won't take effect until the next configuration request packet is received.

Response: If no errors are detected, the DP5 will respond with ACK OK packet. If the data byte is outside the range of 0-5, then a BAD PARAMETER error is returned.

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of ACK OK.

4.1.30 Request packet: "Select RS232 Baud Rate"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7	8
Value	**0xF5**	**0xFA**	**0xF0**	**0x13**	**0**	**1**	**Baud Rate**	**varies**	

Purpose: This selects the RS232 baud rate. The default is 115,200 baud. 57,600 baud or 19,200 baud can be selected; however, these setting are incompatible with the Ethernet stack. Therefore, **selecting 57,600 baud or 19,200 baud will disable the Ethernet interface**. When this value is changed, the new setting will not take effect until the next power cycle. The interpretation of the baud rate value ('setting') is:

0 => 115,200 baud;
1 => 57,600 baud;
2 => 19,200 baud;

Response: If no errors are detected, the DP5 will respond with ACK OK packet. If the data byte is outside the range of 0-2, then a BAD PARAMETER error is returned.

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of ACK OK.

4.1.31 Request packet: "Set HV Calibration"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6-7	8	9
Value	**0xF5**	**0xFA**	**0xF0**	**0x14**	**0**	**2**	**Gain, Offset**	**varies**	

Purpose: This commands the DP5 [PX5] to set the non-volatile gain and offset calibration values for the PC5 [PX5] HV DAC. These values can be read back via the "Request Diagnostic Packet" Request Packet.

Gain: If gain setting is 0-127: Correction = (1024+setting)/1024 (i.e. 100% to 112.4%)
If gain setting is 128-255: Correction = (768+setting)/1024 (i.e. 87.5% to 99.9%)
Offset: The offset setting is a signed byte, with a scale factor of 366mV/LSB.
Calibrated ADC Value = (Raw ADC Value * Gain) + Offset.

Response: If no errors are detected, the DP5 [PX5] will respond with ACK OK packet.

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of ACK OK.

4.1.32 Request packet: "Set 1.6uS PZ Correction" [PX5 only]

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7	8
Value	**0xF5**	**0xFA**	**0xF0**	**0x15**	**0**	**1**	**PZ Corr**	**varies**	

Purpose: The PX5 has two software selectable pole-zero settings; 1.6uS and 3.2uS. This commands the PX5 to set the non-volatile correction factor for the 1.6uS Pole-Zero. This allows digital fine-tuning of the analog PZ time constant. (This calibration value is set at the factory and shouldn't need to be changed.) This value can be read back via the "Request Diagnostic Packet" Request Packet. The interpretation of the PZ Correction value ('setting') is:

Setting of 0-127: the correction factor is (1024+setting)/1024 (i.e. 100% to 112.4%)
Setting of 128-255: the correction factor is (768+setting)/1024 (i.e. 87.5% to 99.9%)
Examples: setting = 0 -> 100% correction factor;
setting =10 -> 101% correction factor;
setting =250-> 99.4% correction factor

Note: This command won't take effect until the next configuration request packet is received.

Response: If no errors are detected, the PX5 will respond with ACK OK packet.

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of ACK OK.

4.1.33 Request packet: "Interface Keep-alive – Allow Sharing"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	**0xF5**	**0xFA**	**0xF0**	**0x20**	**0**	**0**	**0xFD**	**0x01**

Purpose: This Request Packet is used to keep the DP5's UDP socket bound to the IP address and source port of the computer connected to it. The UDP socket resets itself after 4-5 seconds of inactivity (after which any IP address can connect to it), so one of the "Interface Keep-alive" Request Packets should be sent periodically if there is no other periodic traffic, or if the periodic traffic is less frequent than ~2-3 seconds. [If the host wishes to change the Keep-alive status (sharing/no sharing/locked), then a Keep-alive packet should be sent with the new status, even if there is periodic traffic keeping the socket bound.]

The "Allow Sharing" variant of the "Interface Keep-alive" Request Packet tells the DP5 to return the "Sharing Allowed" status in the Netfinder device discovery packet. It also commands the DP5 to accept "Interface Sharing Request" packets on the Netfinder socket, and to return an "Ethernet Sharing Request" ACK packet via the normal UDP socket if such a packet is received. [This ACK packet will replace an OK ACK packet at the next opportunity.]

[As of FW6.01, this Request Packet only applies to Ethernet. In future firmware, it may be expanded to include the other interfaces.]

Response: If no errors are detected, the DP5 will respond with ACK OK packet.

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of ACK OK.

4.1.34 Request packet: "Interface Keep-alive – No Sharing"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	**0xF5**	**0xFA**	**0xF0**	**0x21**	**0**	**0**	**0xFD**	**0x00**

Purpose: This Request Packet is used to keep the DP5's UDP socket bound to the IP address and source port of the computer connected to it. The UDP socket resets itself after 4-5 seconds of inactivity (after which any IP address can connect to it), so one of the "Interface Keep-alive" Request Packets should be sent periodically if there is no other periodic traffic, or if the periodic traffic is less frequent than ~2-3 seconds. [If the host wishes to change the Keep-alive status (sharing/no sharing/locked), then a Keep-alive packet should be sent with the new status, even if there is periodic traffic keeping the socket bound.]

The "No Sharing" variant of the "Interface Keep-alive" Request Packet tells the DP5 to return the "No Sharing" status in the Netfinder device discovery packet. It also commands the DP5 to ignore "Interface Sharing Request" packets on the Netfinder socket.

[As of FW6.01, this Request Packet only applies to Ethernet. In future firmware, it may be expanded to include the other interfaces.]

Response: If no errors are detected, the DP5 will respond with ACK OK packet.

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of ACK OK.

4.1.35 Request packet: "Interface Keep-alive – Lock"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	**0xF5**	**0xFA**	**0xF0**	**0x22**	**0**	**0**	**0xFC**	**0xFF**

Purpose: This Request Packet is used to keep the DP5's UDP socket bound to the IP address and source port of the computer connected to it. Normally, the UDP socket resets itself after 4-5 seconds of inactivity (after which any IP address can connect to it), but it will not automatically reset itself after receiving this Request Packet (so periodic transfers are no longer required to keep the socket bound.) [If the host wishes to change the Keep-alive status (sharing/no sharing/locked), then a Keep-alive packet should be sent with the new status]

The "Lock" variant of the "Interface Keep-alive" Request Packet tells the DP5 to return the "Lock" status in the Netfinder device discovery packet. It also commands the DP5 to ignore "Interface Sharing Request" packets on the Netfinder socket, and to not automatically reset the socket. Care must be taken in using this; the UDP socket binds to the IP Address and source port of the host, so the DP5 will then only communicate with the same IP & source port. The host IP address may change if the host uses a dynamic IP address; the source port may change if the host wishes to communicate with multiple DP5s simultaneously, or doesn't set the source port explicitly. If the host loses the ability to communicate with the socket, then the power will have to be cycled on the DP5 to reset the socket.

[As of FW6.01, this Request Packet only applies to Ethernet. In future firmware, it may be expanded to include the other interfaces.]

Response: If no errors are detected, the DP5 will respond with ACK OK packet.

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of ACK OK.

4.1.36 Request packet: "Comm test - Request ACK packet"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	**0xF5**	**0xFA**	**0xF1**	**0..15**	**0**	**0**	**varies**	

Purpose: This Request Packet is handy for software developers – it requests that a specific ACK packet be returned in response to the request. (It's useful for testing application error handling – this is much easier than trying to force the various error conditions.) The value of PID2 is used as PID2 in the ACK packet; i.e. sending the Request Packet PID1=0xF1/**PID2=4** will yield a 'Checksum Error" ACK packet (PID1=0xFF/**PID2=4**).

Response: An ACK packet, with PID2 equal to the PID2 of the Request Packet. This allows all types of ACK packets to be returned.

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of the requested ACK type.

4.1.37 Request packet: "Comm test - Echo packet"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	DATA FIELD	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6…5+LEN	6+LEN	7+LEN
Value	**0xF5**	**0xFA**	**0xF1**	**0x7F**	**varies**		**Data to be echoed**	**varies**	

Purpose: This Request Packet is handy for software developers – it requests that the DP5 return a Response Packet, whose length and data field are the same as those of this Request Packet – essentially, it echoes the data field in a Response Packet.

Response: If no errors are detected, the DP5 will respond with "Comm test - Echo packet" Response Packet (PID1=0xF1/PID2=0x7F).

Possible errors: If the request packet is malformed or corrupted: CHKSUM, LEN, or PID error ACK packets can be returned instead of the Response Packet.

4.2 Response Packets

Table 2

Description	Support*	PID1	PID2	LEN MSB	LEN LSB	Data (optional)
Status Packet	ALL	0x80	1	0	0x40	See description
256-channel Spectrum	ALL	0x81	1	0x03	0	See description
256-channel Spectrum + status	ALL	0x81	2	0x03	0x40	See description
512-channel Spectrum	ALL	0x81	3	0x06	0	See description
512-channel Spectrum + status	ALL	0x81	4	0x06	0x40	See description
1024-channel Spectrum	ALL	0x81	5	0x0C	0	See description
1024-channel Spectrum + status	ALL	0x81	6	0x0C	0x40	See description
2048-channel Spectrum	ALL	0x81	7	0x18	0	See description
2048-channel Spectrum + status	ALL	0x81	8	0x18	0x40	See description
4096-channel Spectrum	ALL	0x81	9	0x30	0	See description
4096-channel Spectrum + status	ALL	0x81	0x0A	0x30	0x40	See description
8192-channel Spectrum	ALL	0x81	0x0B	0x60	0	See description
8192-channel Spectrum + status	ALL	0x81	0x0C	0x60	0x40	See description
2048-byte scope packet	ALL	0x82	1	8	0	See description
512-byte misc data packet	ALL	0x82	2	2	0	See description
2048-byte scope packet w/ overflow	ALL	0x82	3	8	0	See description
Ethernet settings	ALL	0x82	4	0	0x17	See description
Diagnostic data	ALL	0x82	5	1	0	See description
Hardware description		0x82	6			
Configuration readback packet	6.01	0x82	7	varies		
Netfinder packet	6.01	0x82	8	varies		
I^2C Read Data	6.01	0x82	9	varies		See description
64-byte SCA packet	ALL	0x83	1	0	0x40	See description
Comm test - Echo packet	ALL	0x8F	0x7F	varies		Echo data

* Initial firmware release which supports this packet type

4.2.1 Response packet: "Status Packet"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data Field	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6-69	70	71
Value	0xF5	0xFA	**0x80**	**1**	0	0x40	Status data	varies	

Purpose: This response packet is returned in response to a 'Request Status Packet' Request Packet.

Data Format: The data field contains 64 bytes of status data. The format is given below. (This format also applies to the 64-byte status field in the "Spectrum plus Status" Response Packets.)

Offset	Data bits	Allowed value	Description	Notes
0	D7-D0	0-255	Fast count LSB	
1	D7-D0	0-255	Fast count byte 2	
2	D7-D0	0-255	Fast count byte 3	
3	D7-D0	0-255	Fast count MSB	
4	D7-D0	0-255	Slow count LSB	'Slow counts' are counts from the slow (shaped) channel. Any event that is counted in the spectrum is also counted here.
5	D7-D0	0-255	Slow count byte 2	
6	D7-D0	0-255	Slow count byte 3	
7	D7-D0	0-255	Slow count MSB	
8	D7-D0	0-255	General Purpose Counter LSB	The G.P. Counter can be configured to count a number of different events.
9	D7-D0	0-255	G. P. Counter byte 2	
10	D7-D0	0-255	G. P. Counter byte 3	
11	D7-D0	0-255	G. P. Counter MSB	
12	D7-D0	0-99	Acc. Time (0-99, 1mS/count)	
13	D7-D0	0-255	Acc. Time LSB, 100mS/count	
14	D7-D0	0-255	Acc. Time byte 2	
15	D7-D0	0-255	Acc. Time MSB	
16	D7-D0	0-255	Livetime LSB (1mS/count)	FW6.01: 'Livetime' under development
17	D7-D0	0-255	Livetime byte 2	
18	D7-D0	0-255	Livetime byte 3	
19	D7-D0	0-255	Livetime MSB	
20	D7-D0	0-255	Realtime LSB (1mS/count)	FW6.01 and later
21	D7-D0	0-255	Realtime byte 2	
22	D7-D0	0-255	Realtime byte 3	
23	D7-D0	0-255	Realtime MSB	
24	D7-D4	6-15	Firmware version, major	
	D3-D0	0-15	Firmware version, minor	
25	D7-D4	5-15	FPGA version, major	
	D3-D0	0-15	FPGA version, minor	
26	D7-D0	0-255	Serial Number LSB	
27	D7-D0	0-255	Serial Number byte 2	
28	D7-D0	0-255	Serial Number byte 3	
29	D7-D0	0-255	Serial Number MSB	
30	D7-D0	0-255	HV MSB	DP5G: N/A

31	D7-D0	0-255	HV LSB (signed integer, 0.5V/count)	
32	D7-D4	0	(unused)	
	D3-D0	0-15	Detector temperature MSB	DP5G: N/A
33	D7-D0	0-255	Detector temperature LSB (0.1 degree Kelvin/count)	
34	D7-D0	-128…+127	Board temp (1 °C/count, signed)	
35	D7	0	Preset real time not reached (or not enabled)	
		1	Preset real time reached, MCA stopped	
	D6	0	Auto Fast Threshold not locked	
		1	Auto Fast Threshold locked	
	D5	0	MCA disabled	
		1	MCA enabled	
	D4	0	Preset count not reached	
		1	Preset Count reached, MCA stopped	
	D3	0	TBD (PS status?)	
	D2	0	Oscilloscope data not ready	
		1	Oscilloscope data ready	
	D1	0	Unit has not been configured	
		1	Unit is configured	
	D0	0	TBD	
36	D7	0	Auto Input Offset locked	
		1	Auto Input Offset searching	
	D6	0	MCS not finished	
		1	MCS finished	
	D5	0	No reboot	FW6.05 and later
		1	This is first status packet since reboot occurred	
	D4-D2	0	TBD	
	D1	0	FPGA clock = 20MHz	(formerly status of MCA RAM test, which is now in Diagnostic Packet)
		1	FPGA clock = 80MHz	
	D0	0	FPGA clock was explicitly set with 'CLCK=20' or 'CLCK=80'	
		1	'CLCK=AUTO' was used to select FPGA clock	
37	D7-D4	0	TBD	
	D3-D0	0-15	Firmware Build Number	FW6.06 and later
38	D7	0	PC5 not detected at power-up	DP5G: PC5G not detected PX5: N/A
		1	PC5 detected at power-up	DP5G: PC5G detected PX5: N/A
	D6	0	PC5 HV polarity = negative	DP5G, PX5: N/A
		1	PC5 HV polarity = positive	DP5G, PX5: N/A
	D5	0	PC5 preamp supply = +/- 5V	DP5G, PX5: N/A
		1	PC5 preamp supply = +/- 8.5V	DP5G, PX5: N/A

	D4-D0	0	TBD	
39	D7-D0	0	Device ID: DP5	
		1	Device ID: PX5	
		2	Device ID: DP5G	
40-63	D7-D0	0	N/A (Currently unused)	

4.2.2 Response packet: "256-channel spectrum"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data Field	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6-773	774	775
Value	0xF5	0xFA	**0x81**	**0x01**	3	0	Spectrum Data	varies	

4.2.3 Response packet: "256-channel spectrum plus Status"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data Field		CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6-773	774-837	838	839
Value	0xF5	0xFA	**0x81**	**0x02**	3	0x40	Spectrum Data	Status Data	varies	

4.2.4 Response packet: "512-channel spectrum"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data Field	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6-1541	1542	1543
Value	0xF5	0xFA	**0x81**	**0x03**	6	0	Spectrum Data	varies	

4.2.5 Response packet: "512-channel spectrum plus Status"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data Field		CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6-1541	1542-1605	1606	1607
Value	0xF5	0xFA	**0x81**	**0x04**	6	0x40	Spectrum Data	Status Data	varies	

4.2.6 Response packet: "1024-channel spectrum"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data Field	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6-3077	3078	3079
Value	0xF5	0xFA	**0x81**	**0x05**	0x0C	0	Spectrum Data	varies	

4.2.7 Response packet: "1024-channel spectrum plus Status"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data Field		CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6-3077	3078-3141	3142	3143
Value	0xF5	0xFA	**0x81**	**0x06**	0x0C	0x40	Spectrum Data	Status Data	varies	

4.2.8 Response packet: "2048-channel spectrum"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data Field	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6-6149	6150	6151
Value	0xF5	0xFA	**0x81**	**0x07**	0x18	0	Spectrum Data	varies	

4.2.9 Response packet: "2048-channel spectrum plus Status"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data Field	CHKSUM MSB	CHKSUM LSB

Offset	0	1	2	3	4	5	6-6149	6150-6213	6214	6215
Value	0xF5	0xFA	**0x81**	**0x08**	0x18	0x40	Spectrum Data	Status Data	varies	

4.2.10 Response packet: "4096-channel spectrum"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data Field	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6-12293	12294	12295
Value	0xF5	0xFA	**0x81**	**0x09**	0x30	0	Spectrum Data	varies	

4.2.11 Response packet: "4096-channel spectrum plus Status"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data Field		CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6-12293	12294-12357	12358	12359
Value	0xF5	0xFA	**0x81**	**0x0A**	0x30	0x40	Spectrum Data	Status Data	varies	

4.2.12 Response packet: "8192-channel spectrum"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data Field	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6-24581	24582	24583
Value	0xF5	0xFA	**0x81**	**0x0B**	0x60	0	Spectrum Data	varies	

4.2.13 Response packet: "8192-channel spectrum plus Status"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data Field		CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6-24581	24582-24645	24646	24647
Value	0xF5	0xFA	**0x81**	**0x0C**	0x60	0x40	Spectrum Data	Status Data	varies	

Purpose: These response packets are sent by the DP5 in response to a request for spectrum data – the specific response packet returned depends on how many channels the DP5's MCA is configured for, and whether the request was for spectrum data only, or spectrum plus status data.

Data Format: Spectrum data is 3 bytes/channel, LSB to MSB, starting with channel 0, up to the maximum channel number. (i.e. in 256 channel mode, the channel data goes from channel 0 to channel 255.) See table XX. The status data follows the spectrum data in the Response Packet data field and is 64 bytes in length. The format is the same as is specified for the 'Status' response packet.

4.2.14 Response packet: "2048-byte Scope Packet"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data Field	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6-2053	2054	2055
Value	0xF5	0xFA	**0x82**	**1**	8	0	Scope data	Varies	

4.2.15 Response packet: "2048-byte Scope Packet w/ Overflow"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data Field	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6-2053	2054	2055
Value	0xF5	0xFA	**0x82**	**3**	8	0	Scope data	Varies	

Purpose: These response packets are returned in response to a 'Request Digital Scope Data' packet (PID1=3, PID2=1). Only one is returned – the second one (PID2=3) is returned if the Scope Gain is set to 4 or 16, and the signal strayed outside the gain range during the scope acquisition. Otherwise, the first one (PID2=1) is returned. In either case, the data is identical. (This second type is used for noise measurements.) [See the ASCII commands DACO, SCOE, SCOT & SCOG for information on configuring the Digital Scope].

Data Format: The data field contains 2048 bytes of scope data, where each byte is a single 8-bit sample from the scope. The Digital Scope captures the most significant 8 bits of digital data that is being output on the output DAC.

The timebase for the scope is dependent on the FPGA clock and the peaking time selected, as listed below. (The timebase listed is the interval between samples.)

FPGA Clock=20MHz		FPGA Clock=80MHz	
Peaking Time	**Scope Timebase**	**Peaking Time**	**Scope Timebase**
0.800 - 6.40uS	50nS	0.200 - 1.600uS	12.5nS
6.60 - 12.80uS	100nS	1.650 - 3.200uS	25nS
13.20 - 25.60uS	200nS	3.30 - 6.40uS	50nS
26.40 - 51.20uS	400nS	6.60 - 12.80uS	100nS
52.80 - 102.4uS	800nS	13.20 - 25.60uS	200nS

4.2.16 Response packet: "512-byte Misc Data Packet"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data Field	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6-517	518	519
Value	0xF5	0xFA	**0x82**	**2**	2	0	Misc data	Varies	

Purpose: This response packet is returned in response to a 'Request Misc Data' packet (PID1=3, PID2=2). [See the 'Write Misc Data' Request Packet for information on writing the Misc Data.]

Data Format: The data field contains 512 bytes, which is the Misc Data programmed via a 'Write Misc Data' request packet. The data can be binary, ASCII, etc. This is stored in non-volatile memory, so it remains after the power is cycled.

4.2.17 Response packet: "Ethernet Settings"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data Field	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6-30	31	32
Value	0xF5	0xFA	**0x82**	**4**	0	0x19	Ethernet settings	Varies	

Purpose: This response packet is returned in response to a 'Request Ethernet Settings' packet (PID1=3, PID2=4).

Data Format: The data field contains 23 bytes of Ethernet Settings, in the following format:

Offset	Value
0	0 = Use fixed IP address; 0xFF = Get IP address from DHCP server
1-4	IP Address (MSB…LSB)
5-8	IP Subnet Mask (MSB…LSB)
9-12	IP Gateway Address (MSB…LSB)
13-16	Reserved
17-18	Port (MSB,LSB: fixed at 10001 for now)
19-24	DP5 MAC Address

4.2.18 Response packet: "Diagnostic Data"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data Field	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6-261	262	263
Value	0xF5	0xFA	**0x82**	**5**	1	0	Diagnostic Data	Varies	

Purpose: This response packet is returned in response to a 'Request Diagnostic Data' Request Packet (PID1=3, PID2=5).

Data Format: The data field contains 256 bytes of various types of diagnostic data, in the following format: [Note: PX5 & DP5G bytes 5-38 are different!]

Offset	Bits	Value	Comment
0	D7-D4	Firmware (major)	6.0 for this release
	D3-D0	Firmware (minor)	
1	D7-D4	FPGA (major)	5.07 for this release;
	D3-D0	FPGA (minor)	0.0 indicates FPGA didn't initialize
2-4	all	SRAM error address (MSB…LSB)	0xFFFFFF indicates no error
DP5, Bytes 5-38			
5	D3-D0	DP5 board temp (raw), MSB	Temp(°C) = (MSB*256 + LSB) * 349.7 – 271.3
6	D7-D0	DP5 board temp (raw), LSB	[Offset 180 is the 8-bit signed offset cal for this]
7	D7-D0	TBD	
8	D7-D0	TBD	
9	D1-D0	Power Input, MSB	Input(V) = (MSB*256 + LSB) / 167.5
10	D7-D0	Power Input, LSB	
11	D1-D0	3.3V supply, MSB	3.3V(V) = (MSB*256 + LSB) / 254.3
12	D7-D0	3.3V supply, LSB	
13	D1-D0	2.5V supply, MSB	2.5V(V) = (MSB*256 + LSB) / 335.7
14	D7-D0	2.5V supply, LSB	
15	D1-D0	1.2V supply, MSB	1.2V(V) = (MSB*256 + LSB) / 419.7
16	D7-D0	1.2V supply, LSB	
17	D1-D0	5.5V supply, MSB	5.5V(V) = (MSB*256 + LSB) / 150.7
18	D7-D0	5.5V supply, LSB	(This is called '5.5MON' and is used in

			the -5.5V monitor below)
19	D1-D0	-5.5V supply, MSB	-5.5(V) = (MSB*256 + LSB) / 135.5 - (2.101 * 5.5MON)
20	D7-D0	-5.5V supply, LSB	
21	D1-D0	AN_IN, MSB	AN_IN(V) = (MSB*256 + LSB) / 419.7 (AN_IN is on J5.11)
22	D7-D0	AN_IN, LSB	
23	D1-D0	VREF_IN, MSB	VREF_IN(V) = (MSB*256 + LSB) / 419.7 (VREF_IN is a test point)
24	D7-D0	VREF_IN, LSB	
Offsets 25-38 will be zero if a PC5 is not detected			
25	D3-D0	PC5 HV Mon, MSB	HV (V) = (MSB*256 + LSB) / 2.731
26	D7-D0	PC5 HV Mon, LSB	
27	D3-D0	PC5 Det Temp, MSB	Det Temp (K) = (MSB*256 + LSB) / 13.65
28	D7-D0	PC5 Det Temp, LSB	
29	D3-D0	PC5 +PA supply, MSB	+PA (V) = (MSB*256 + LSB) / 455.1
30	D7-D0	PC5 +PA supply, LSB	
31-34	D7-D0	PC5 S/N, LSB…MSB	
35	D3-D0	DCAL, MSB	DCAL (V) = (MSB*256 + LSB) / 1365
36	D7-D0	DCAL, LSB	
37	D7	PC5 HV polarity	0 = -HV, 1=+HV
	D6	PC5 PA supply	0 = 5V preamp, 1 = 8.5V preamp
	D5-D0	N/A	
38	D7-D4	N/A	
	D3	PC5 9V supply	0 = OFF, 1 = ON
	D2	PC5 preamp supplies	0 = OFF, 1 = ON
	D1	PC5 HV supply	0 = OFF, 1 = ON
	D0	PC5 TEC supply	0 = OFF, 1 = ON
PX5, Bytes 5-38			
5	D3-D0	9V supply, MSB	9V (V) = (MSB*256 + LSB) / 304.1
6	D7-D0	9V supply, LSB	
7	D7-D0	3.3V supply, MSB	3.3V (V) = (MSB*256 + LSB) / 827.5
8	D7-D0	3.3V supply, LSB	
9	D1-D0	2.5V supply, MSB	2.5V (V) = (MSB*256 + LSB) / 1093
10	D7-D0	2.5V supply, LSB	
11	D1-D0	1.2V supply, MSB	1.2 (V) = (MSB*256 + LSB) / 1365

12	D7-D0	1.2V supply, LSB	
13	D1-D0	5V supply, MSB	5V (V) = (MSB*256 + LSB) / 545
14	D7-D0	5V supply, LSB	
15	D1-D0	-5V supply, MSB	-5V (V) = (MSB*256 + LSB) / 170.5 – 21.03
16	D7-D0	-5V supply, LSB	
17	D1-D0	+PA supply, MSB	+PA (V) = (MSB*256 + LSB) / 321.6
18	D7-D0	+PA supply, LSB	
19	D1-D0	-PA supply, MSB	-PA (V) = (MSB*256 + LSB) / 170.5 – 21.03
20	D7-D0	-PA supply, LSB	
21	D1-D0	TEC supply, MSB	TEC (V) = (MSB*256 + LSB) / 758.5
22	D7-D0	TEC supply, LSB	
23	D1-D0	HV supply, MSB	HV (V) = (MSB*256 + LSB) / 2.731
24	D7-D0	HV supply, LSB	
25	D3-D0	Detector Temperature, MSB	Det Temp (K) = (MSB*256 + LSB) / 13.65
26	D7-D0	Detector Temperature, LSB	
27	D1-D0	Board temperature (raw), MSB	Temp(°C) = (MSB*256 + LSB) * 1.024 – 271.3
28	D7-D0	Board temperature (raw), LSB	
29-38 TBD			
DP5G Bytes 5-38			
5-38	TBD		
Offsets 39-230 contain the Boot Record - Select items are listed below			
167-168		HV setting	Signed INT, MSB…LSB (in Volts)
169-170		TEC setting	Unsigned INT, MSB…LSB (in K)
171-172		Input Offset	Signed INT, MSB…LSB (in mV) [-2048 indicates default setting]
173-176		Preset Time	Unsigned LONG, MSB…LSB (0.1s/lsb)
177-178		DP5 DCAL	Unsigned INT, MSB…LSB (732uV/lsb) [PC5 DCAL takes precedence, if present]
179		PZ Correction	See the "Set PZ Correction" Request Packet for definition

180		uC Temp Calibration	See the "Set uC Temperature Calibration" Request Packet for definition
181		ADC Gain Calibration	See the "Set ADC Calibration" Request Packet for definitions
182		ADC Offset Calibration	
187		Highpass time constant	See the "Select High-Pass Time Constant" Request Packet for definition
188 [DP5] 189 [DP5G] 194 [PX5]		RS232 Baud rate	See the "Select RS232 Baud Rate" Request packet for definition
189 [DP5] 195 [PX5]		HV gain calibration	See the "Set HV Calibration" Request Packet for definition
190 [DP5] 196 [PX5]		HV offset calibration	
197 [PX5]		PZ Correction (1.6uS)	See the "Set 1.6uS PZ Correction" Request Packet for definition

4.2.19 Response packet: "Configuration Readback"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data Field	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6…5+LEN	6+LEN	7+LEN
Value	0xF5	0xFA	**0x82**	**7**	varies		Configuration readback (ASCII)	varies	

Purpose: This response packet is returned in response to a 'Configuration Readback' Request Packet (PID1=3, PID2=5). The data field contains ASCII commands and parameters, separated by semicolons.

Data Format: The data field will contain the current ASCII configuration – it will have all the commands that were sent with the "Configuration Readback" Request Packet, except the parameters will have been replaced with the actual settings in use by the DP5. See the "Text Configuration" Request Packet, "Configuration Readback" Request packet, and the section on ASCII Commands for more details.

4.2.20 Response packet: "Netfinder Packet"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	DATA FIELD	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6…5+LEN	6+LEN	7+LEN
Value	**0xF5**	**0xFA**	**0x82**	**8**	**varies**		**Netfinder data**	**varies**	

Purpose: This is used to return the 'Netfinder' packet via the normal communications link. [Normally, the Netfinder data is returned by a dedicated TCP/IP socket.]

Data Format: The data field is formatted the same as an independent Netfinder packet, as documented in section 3.2.3.

4.2.21 Response packet: "I²C Read Data"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	DATA FIELD	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6…5+LEN	6+LEN	7+LEN
Value	**0xF5**	**0xFA**	**0x82**	**9**	**varies**		**I²C read data**	**varies**	

Purpose: This is used to return the data read by the 'I²C Transfer' Request Packet. If the Number of Bytes to Read ('NOBR') in the 'I²C Transfer' Request Packet is nonzero, and the I²C transfer was successful, then the 'I²C Read Data' Response Packet is returned, with the data field containing the bytes read by the transfer. [Note that if NOBR = 0, the DP5 will return an ACK OK packet rather than an 'I²C Read Data' packet with LEN=0.]

Data Format: The LEN value will be the same as NOBR, and the data field will contain the bytes read, in the order that they were read.

4.2.22 Response packet: "64-byte SCA Packet"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	Data Field	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6-69	70	71
Value	0xF5	0xFA	**0x83**	**1**	0	0x40	SCA data	Varies	

Purpose: This response packet is returned in response to a 'Request 32-bit SCA counters' Request Packet (PID1=4, PID2=1, 2 or 3.)

Data Format: The data field contains 64 bytes, which are read from the 32-bit SCA counter latches (i.e. there are 4 bytes per SCA counter.) The format for the data field is given below:

Offset	Value
0-3	SCA1 counter (LSB…MSB)
4-7	SCA2 counter (LSB…MSB)
8-11	SCA3 counter (LSB…MSB)
12-15	SCA4 counter (LSB…MSB)
16-19	SCA5 counter (LSB…MSB)
20-23	SCA6 counter (LSB…MSB)
24-27	SCA7 counter (LSB…MSB)
28-31	SCA8 counter (LSB…MSB)
32-35	SCA9 counter (LSB…MSB)
36-39	SCA10 counter (LSB…MSB)
40-43	SCA11 counter (LSB…MSB)
44-47	SCA12 counter (LSB…MSB)
48-51	SCA13 counter (LSB…MSB)
52-55	SCA14 counter (LSB…MSB)
56-59	SCA15 counter (LSB…MSB)
60-63	SCA16 counter (LSB…MSB)

4.2.23 Response packet: "Comm test - Echo packet"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	DATA FIELD	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6…5+LEN	6+LEN	7+LEN
Value	**0xF5**	**0xFA**	**0x8F**	**0x7F**	**varies**		**Echo data**	**varies**	

Purpose: This Response Packet is sent in response to the Request Packet of the same name. The LEN & DATA fields are the same as in the Request Packet.

Data Format: Whatever data is in the data field of the Request Packet is copied, byte-for-byte, into the data field of the Response packet.

4.3 Acknowledge Packets

Table 3

Description	Support*	PID1	PID2	LEN MSB	LEN LSB	Data (optional)
OK	ALL	0xFF	0	0	0	
Sync error	ALL	0xFF	1	0	0	
PID error	ALL	0xFF	2	0	0	
LEN error	ALL	0xFF	3	0	0	
Checksum error	ALL	0xFF	4	0	0	
Bad parameter	ALL	0xFF	5	varies		ASCII command
Bad hex record (structure/chksum)	ALL	0xFF	6	0	0	
Unrecognized command	ALL	0xFF	7	varies		ASCII command
FPGA error (not initialized)	ALL	0xFF	8	0	0	
CP2201 not found	ALL	0xFF	9	0	0	
Scope data not available (not triggered)	ALL	0xFF	0x0A	0	0	
PC5 not present	ALL	0xFF	0x0B	varies		ASCII command
OK + Interface sharing request	ALL	0xFF	0x0C	0	0	
Busy - another interface is in use		0xFF	0x0D	0	0	
I2C error	6.01	0xFF	0x0E	0	0	

* Initial firmware release which supports this packet type

4.3.1 Acknowledge packet: "OK"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	0xF5	0xFA	**0xFF**	**0**	0	0	0xFD	0x12

Purpose: This ACK packet indicates that the Request Packet was received without error, that its contents were recognized, and that the requested action was carried out. This ACK packet is only returned by Request Packets which do not request data.

4.3.2 Acknowledge packet: "OK, with Interface Sharing Request"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	0xF5	0xFA	**0xFF**	**0x0C**	0	0	0xFD	0x06

Purpose: This ACK packet has the same meaning as the "OK" ACK packet. In addition, another computer or interface is requesting that activity be paused so that it can gain access to the DP5. To allow it access, the application should stop sending traffic to the DP5. To maintain control (and deny it access), continue sending Request Packets. (Send 'Interface keep-alive' packets periodically, if no other periodic traffic is occurring. See the Interface section for more details.)

4.3.3 Acknowledge packet: "Sync Error"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	0xF5	0xFA	**0xFF**	**1**	0	0	0xFD	0x11

Purpose: This ACK packet indicates that the sync bytes in Request Packet were not correct, and therefore, the Request Packet was rejected.

Note: This error condition can only be generated via USB or Ethernet; RS232 uses the sync bytes to locate a packet in the RS232 data stream. In the absence of the sync byte pattern, no Request Packet is detected and thus no ACK packet can be returned.

4.3.4 Acknowledge packet: "PID Error"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	0xF5	0xFA	**0xFF**	**2**	0	0	0xFD	0x10

Purpose: This ACK packet indicates that PID1 & PID2 combination is not recognized as a valid Request Packet, and therefore, the Request Packet was rejected.

4.3.5 Acknowledge packet: "LEN Error"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	0xF5	0xFA	**0xFF**	**3**	0	0	0xFD	0x0F

Purpose: This ACK packet indicates that the LEN field of the Request Packet was not consistent with Request Packet type defined by the PID1 & PID2 combination. It is not recognized as a valid Request Packet, and therefore, the Request Packet was rejected.

4.3.6 Acknowledge packet: "Checksum Error"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	0xF5	0xFA	**0xFF**	**4**	0	0	0xFD	0x0E

Purpose: This ACK packet indicates that the checksum of the Request Packet was incorrect, and therefore, the Request Packet was rejected.

4.3.7 Acknowledge packet: "Bad Parameter"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	DATA	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6...5+LEN	6+LEN	7+LEN
Value	0xF5	0xFA	**0xFF**	**5**	0	varies	ASCII Command Echo	varies	

4.3.8 Acknowledge packet: "Unrecognized Command"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	DATA	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6...5+LEN	6+LEN	7+LEN
Value	0xF5	0xFA	**0xFF**	**7**	0	varies	ASCII Command Echo	varies	

4.3.9 Acknowledge packet: "PC5 Not Present"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	DATA	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6...5+LEN	6+LEN	7+LEN
Value	0xF5	0xFA	**0xFF**	**0x0B**	0	varies	ASCII Command Echo	varies	

Purpose: These ACK packets are in response to ASCII command errors – the data field will contain the ASCII command and parameter which caused the error. "Bad Parameter" means that the parameter isn't recognized or exceeds the range of the command. "Unrecognized Command" means that the 4-character command isn't recognized. "PC5 Not Present" is returned if a PC5 is not mated to the DP5, and a command requiring a PC5 is sent. (i.e. "HVSE", Set High Voltage.) [A 'Bad Parameter' ACK packet may also be returned for a malformed I2C Request Packet, in which case LEN=0.]

If an incomplete or garbled command is returned in the data field, it may mean that the ASCII Configuration Packet has structural issues. (Disallowed whitespace, missing semicolon, etc.)

4.3.10 Acknowledge packet: "Bad Hex Record"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	0xF5	0xFA	**0xFF**	**6**	0	0	0xFD	0x0C

Purpose: This ACK packet applies to microcontroller or FPGA upload packets (PID1=0x30, PID2=2 or 7.) It means that the hex record contained in the data field of the Request Packet had a checksum or other structural error.

4.3.11 Acknowledge packet: "FPGA Error"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	0xF5	0xFA	**0xFF**	**8**	0	0	0xFD	0x0A

Purpose: This ACK packet indicates that the FPGA failed initialization, and is returned in response to "Request Spectrum" (PID1=2) packets.

4.3.12 Acknowledge packet: "CP2201 Not Found"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	0xF5	0xFA	**0xFF**	**9**	0	0	0xFD	0x09

Purpose: This ACK packet indicates that the "Set Ethernet Settings" (PID1=0xF0, PID2=0x11) was received, but an Ethernet controller was not detected on the DP5. [Some early DP5s were manufactured without an Ethernet controller.]

4.3.13 Acknowledge packet: "Scope Data Not Available"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	0xF5	0xFA	**0xFF**	**0x0A**	0	0	0xFD	0x08

Purpose: This ACK packet indicates that a "Send Scope Data" (PID1=3, PID2=1 or 3) was received, but the digital oscilloscope hasn't triggered, so no data is available. [The digital oscilloscope must be armed, and then a trigger must occur for data to be available.]

4.3.14 Acknowledge packet: "I^2C Error"

	SYNC1	SYNC2	PID1	PID2	LEN MSB	LEN LSB	CHKSUM MSB	CHKSUM LSB
Offset	0	1	2	3	4	5	6	7
Value	0xF5	0xFA	**0xFF**	**0x0E**	0	0	0xFD	0x04

Purpose: This ACK packet indicates that the I^2C Bus Master attempted a transfer as described by the 'I^2C Transfer' Request Packet, but no I^2C ACK was detected from an I^2C Slave.

5 ASCII Commands

In the following section, the format is specified for all the ASCII commands recognized by the DP5. These commands are sent to the DP5 in the 'Text Configuration' Request Packet (PID1=0x20, PID2=2). The commands can be sent singly, but most often, they are grouped together into a single Text Configuration packet so that the entire configuration can be sent at once.

Generally, when the entire configuration is sent at once, the first command should be "RESC=Y". This resets the configuration to a known state before applying the configuration commands. This is not automatically done because it may be possible that the required configuration is larger than can be sent in a single 'Text Configuration' Request Packet. If this is the case, then "RESC=Y" and as many commands as will fit should be sent in the first Text Configuration packet, with the remaining commands in later packets. Also, it may be desirable to send individual commands to fine-tune the configuration.

This is discussed in the 'Text Configuration' Request Packet definition, but it's worth mentioning again: all ASCII characters must be upper-case; no whitespace is allowed (i.e. no spaces, tabs, CR-LF, etc.); and the commands are separated by semicolons.

Some commands have dependencies; they require other commands to be listed before them. For example, the Peaking Time command (TPEA) needs to know what FPGA clock to use, so it requires the FPGA Clock (CLCK) command to be listed before it. Likewise, many commands are dependent on the TPEA command, so it must precede them. If a dependent command is specified, that command must occur prior to the command which lists the dependency. The table below specifies the 'Order' for each command – lower Order commands must be sent before higher Order commands.

5.1 Table 4 – ASCII Command Summary

Description	Support*	Command	Parameter	Units	Default	ORDER	Limits
ACK / Don't ACK packets with errors		ACKE	[ON|OF{F}]				
Analog input pos/neg	ALL	AINP	[PO{S}|NE{G}]		NEG		
AUX_OUT selection	ALL	AUO1	[#|ICR|PILEUP|MCSTB|ONESH|DETRES|MCAEN|PEAKH|SCA8]		ICR		1-8 or the exact strings listed
AUX_OUT2 selection	ALL	AUO2	[#|ICR|DIAG|PEAKH|ONESH|RTDOS|RTDREJ|LIVE|VETO]		ICR		1-8 or the exact strings listed
BLR down correction	ALL	BLRD	[#]		0		0-3
BLR mode	ALL	BLRM	[OF{F}|1]		OFF		only '1' for now
BLR up correction	ALL	BLRU	[#]		0		0-3
Turn supplies on/off at power up	ALL; D	BOOT	[ON|OF{F}]				
Select Connector 1 signal	6.03; P, G	CON1	[DAC|AUXOUT1|AUXIN1]		AUXIN1 (DP5G); DAC (PX5)		
Select Connector 2 signal	6.03; P, G	CON2	[AUXOUT2|AUXIN2 GATEH|GATEL]		AUXIN2 (DP5G); AUXOUT2 (PX5)		
20MHz/80MHz	ALL	CLCK	[AU{TO}|20|80]		AUTO ['20' for DP5G]	2	
Non-trapezoidal shaping	ALL	CUSP	[OFF|##]		0% (trapezoid)		-99 to +99% (0%=trapezoid)
DAC offset	ALL	DACF	[{-}###]	[MV]	0mV		-500 to +499mV
DAC output	ALL	DACO	[#|OFF|FAST|SHAPED|INPUT| PEAK]		OFF		1-8
Analog gain index	ALL	GAIA	[##]				DP5: 1-16 PX5: 1-28 DP5G: 1-4
Fine gain	ALL	GAIF	[##.####]			4	0.5-1.9999
Total Gain (analog * fine)	ALL	GAIN	[###.###]			4	DP5: 0.75-150 PX5: 0.75-500 DP5: 1-10
Gate control	ALL; D	GATE	[OF{F}|HI{GH}|LO{W}]		OFF		
G.P. counter edge	ALL	GPED	[RI{SING}|FA{LLING}]		FALLING		
G.P. counter uses GATE	ALL	GPGA	[ON|OF{F}]		ON		
G.P. counter input	ALL	GPIN	[#|AUX1|AUX2|PILEUP|RTDREJ|SCA8|TBD|DETRES|OFF]		AUX1		1-8
G.P. counter cleared with MCA	ALL	GPMC	[ON|OF{F}]		ON		

counters?							
G.P. counter uses MCA_EN?	ALL	GPME	[ON|OF{F}]		ON		
HV set	ALL	HVSE	[{+|-}####|OF{F}]	[V]	OFF		0 to 1499V (+HV); 0 to -1499V (-HV)
Input offset	ALL; D, P	INOF	[{+|-}####|AU{TO}|DE{F}]	[MV]	DEF	AINP (for DEF)	-2047 to +2047mV
Input offset gain	6.06; P	INOG	[LO{W}|HI{GH}|		LOW		
MCA/MCS channels	ALL	MCAC	[256|512|1024|2048| 4096|8192]		1024		
MCA/MCS enable	ALL	MCAE	[ON|OF{F}]		OFF (disabled)		
MCA Source	ALL	MCAS	[NO{RM}|MC{S}|FA{ST}| PU{R}|RT{D}]		NORM	6	
MCS low threshold	6.01	MCSL	[####]	[CH]	0		0-8191
MCS high threshold	6.01	MCSH	[####]	[CH]	8191		0-8191
MCS timebase	ALL	MCST	[###.###]	[S]	0		0.01-655.35S
preamp 8.5/5	ALL; D, P	PAPS	DP5: [8{.5}|5|OF{F}|ON] PX5: [8{.5}|5|OF{F}]	[V]	OFF		
Preamp pole-zero cancellation	6.05; P	PAPZ	[####.#|OFF]	[US]	OFF		34.5-4387uS
Peak detect mode (min/max)	ALL	PDMD	[NORM|MIN]		NORM		
Preset counts, low threshold	6.01	PRCL	[####]	[CH]	0		0-8191
Preset counts, high threshold	6.01	PRCH	[####]	[CH]	8191		0-8191
Preset counts	ALL	PREC	[##########|OF{F}]		OFF		0-4294967295 (2^32 -1)
Preset Real Time	6.01	PRER	[########.###|OF{F}]		OFF		0 – 4,294,967.29 s
Preset time	ALL	PRET	[#######.#|OF{F}]	[S]	OFF		0-99999999.9s
PUR interval, on/off	ALL	PURE	[ON|OF{F}|MAX|###.###]	[US]	OFF	4 (for ###.###)	
Reset Configuration	ALL	RESC	[Y{ES}|NO]			1	
Detector Reset lockout	ALL	RESL	[#####|OF{F}]	[US]	OFF	4	0-65535uS
Custom RTD oneshot delay	ALL	RTDD	[###]			6	1-127 decimated clocks
RTD on/off	ALL	RTDE	[ON|OF{F}]		OFF	5	
RTD sensitivity	ALL	RTDS	[####]	[%]	0		2-1593
RTD threshold	ALL	RTDT	[##.###]	[%]	0		0-49.9
Custom RTD oneshot width	ALL	RTDW	[###]			6	1-127 decimated clocks
SCAx high threshold	ALL	SCAH	[####]		0 (all SCAs)		0-8191

SCA index	ALL	SCAI	[##] (1-16)				1-16
SCAx low theshold	ALL	SCAL	[####]		0 (all SCAs)		0-8191
SCAx output (SCA1-8 only)	ALL	SCAO	[OF{F}|HI{GH}|LO{W}]		OFF (all SCAs)		
SCA pulse width (not indexed - SCA1-8)	ALL	SCAW	[100|1000]	[NS]	100		
Scope trigger edge	ALL	SCOE	[RI{SING}|FA{LLING}]		RISING		
Digital scope gain	ALL	SCOG	[1|4|16]		1		
Scope trigger position	ALL	SCOT	[87|50|12|-25]	[%]	87%		
Set spectrum offset	ALL	SOFF	[OFF|{-}####.###]	[CH]	OFF	after MCAC	-8192 to +8191.75 (8K channels); -4096 to +4095.875 (4K channels)…
TEC set	ALL; D, P	TECS	[###|OF{F}]	[K]	OFF		0-299K
Flat top	ALL	TFLA	[##.###]	[US]	0uS	4	0-51.2uS
Fast threshold	ALL	THFA	[###.###]		0		0-255.937
Slow threshold	ALL	THSL	[##.###]	[%]	0		0-24.9%
LLD threshold	6.01	TLLD	[####|OF{F}]	[CH]	OFF		0-8191
peaking time	ALL	TPEA	[###.###]	[US]	no default	3	0.8-102.4uS (CLK=20MHz); 0.2-25.6uS (80MHz); 0.2-102.4uS (AUTO)
Fast channel peaking time	ALL	TPFA	[50|100|200|400|1600]	[NS]	100 (80MHz); 400 (20MHz)	4	50, 100, 400 (80MHz); 200, 400, 1600 (20MHz)
Test pulser on/off	ALL	TPMO	[OF{F}|+S{NG}|+D{BL}|-S{NG}|-D{BL}]		OFF		
Volume [PX5 only]	ALL; P	VOLU	[OF{F}|ON]		OFF		

*Support: Minimum firmware (FW) version to support command, plus which devices support it;
'D' = DP5; 'P' = PX5, 'G' = DP5G (GammaRad). If none are listed, then all 3 devices support the command

5.1.1 AINP - Set the Input Polarity

Command: AINP
Parameter: [PO{S}|NE{G}]
Units: none
Default: NEG

Dependencies: none
Supported: FW6.00 and later
Related: INOF

Description: AINP selects the polarity of the pulses at the DP5 signal input. An Amptek Si-PIN detector produces negative-going steps; the AINP setting is 'NEG'. An Amptek SDD detector produces positive-going steps; the AINP setting 'POS'. [The AINP command is not related to the voltage polarity of the input; it is used for selecting the polarity of the input pulses.]

Errors: An INVALID PARAMETER error will be returned if the parameter is not one of those listed.

Example: AINP=POS; // Select +input pulses for SDD

5.1.2 AUO1 - Select AUX_OUT1 Signal

Command: AUO1
Parameter: [#|ICR|PILEUP|MCSTB|ONESH| DETRES|MCAEN|PEAKH|SCA8]
Units: none
Default: ICR
Range: 1-8 or the exact strings listed

Dependencies: none
Supported: FW6.00 and later
Related: SCOE, SCOT, CON1 (PX5/DP5G)

Description: AUO1 selects the source for the AUX_OUT1 signal. The selected signal is also used as the trigger source for the digital oscilloscope. The parameter can specify the signal name, or an index (1=ICR… 8=SCA8). [Note that PX5 and DP5G require the 'CON1' command in order to connect the AUXOUT1 signal to the AUX1 connector.]

Errors: An INVALID PARAMETER error will be returned if the parameter is not one of the values specified, or a number in the range specified, in which case the default of ICR will be used. [Most of the commands which have text parameters only require the first two characters; this command requires all characters specified.]

Example:
```
AUO1=PILEUP;        // Set the AUX_OUT1 signal to PILEUP,
                    // and trigger the digital scope on it
```

5.1.3 AUO2 - Select AUX_OUT2 Signal

Command: AUO2
Parameter: [#|ICR|DIAG|PEAKH|ONESH|RTDOS|RTDREJ|LIVE|VETO]
Units: none
Default: ICR
Range: 1-8 or the exact strings listed

Dependencies: none
Supported: FW6.00 and later (FW6.02 and later for string names)
Related: CON2 (PX5/DP5G)

Description: AUO2 selects the source for the AUX_OUT2 signal. The parameter can specify the signal name, or an index (1=ICR… 8=VETO). [Note that PX5 and DP5G require the 'CON2' command in order to connect the AUXOUT2 signal to the AUX2 connector.]

Errors: An INVALID PARAMETER error will be returned if the parameter is not one of the values specified, or a number in the range specified, in which case the default of ICR will be used. [Most of the commands which have text parameters only require the first two characters; this command requires all characters specified.]

Example: AUO2=RTDOS; // Set the AUX_OUT2 signal to RTDOS

5.1.4 BLRD - Select the Baseline Restorer 'Down' Correction

Command: BLRD
Parameter: [#]
Units: none
Default: 0
Range: 0-3

Dependencies: none
Supported: FW6.00 and later
Related: BLRM, BLRU

Description: BLRD selects the size of the BLR 'down' correction. '0' is the smallest (slowest, least aggressive) and '3' is the largest (fastest, most aggressive.)

Errors: An INVALID PARAMETER error will be returned if the parameter is not in the *range* specified.

Example:
```
BLRD=3;          // Select the most aggressive BLR 'down'
                 // correction
```

5.1.5 BLRM - Select the Baseline Restorer Mode

Command: BLRM
Parameter: [OF{F}|1]
Units: none
Default: OFF

Dependencies: none
Supported: FW6.00 and later
Related: BLRU, BLRD

Description: BLRM selects whether the Baseline Restorer is enabled, and if so, which mode it will operate in. In the initial release (FW6.00), only BLR mode 1 is supported; others will be added in the future.

Errors: An INVALID PARAMETER error will be returned if the parameter is not 'OFF' or '1'.

Example: BLRM=1; // Enable the BLR

5.1.6 BLRU - Select the Baseline Restorer 'Up' Correction

Command: BLRU
Parameter: [#]
Units: none
Default: 0
Range: 0-3

Dependencies: none
Supported: FW6.00 and later
Related: BLRM, BLRD

Description: BLRU selects the size of the BLR 'up' correction. '0' is the smallest (slowest, least aggressive) and '3' is the largest (fastest, most aggressive.)

Errors: An INVALID PARAMETER error will be returned if the parameter is not in the *range* specified.

Example: BLRU=1; // Select the 2nd slowest BLR 'up' correction

5.1.7 BOOT - Set Power-on State

[DP5]

Command: BOOT
Parameter: [ON|OF{F}]
Units: none
Default: none

Dependencies: none
Supported: FW6.00 and later; DP5 only [not supported by PX5 or DP5G – for PX5, power button is held in at power-up to start with power supplies on. DP5G/PCG always boots with HV DAC set to its previous state.]
Related: HVSE, PAPS, TECS

Description: BOOT determines whether the PC5 power supplies (HV, TEC & Preamp) will be turned on automatically when power is applied to the DP5/PC5. If this is set to 'ON', then the next time the DP5/PC5 is powered up, the HV, TEC & Preamp supplies will automatically be set to the previously applied settings. The 'BOOT' setting is non-volatile; its setting will be retained and is not reset by the Reset Configuration ("RESC") command. Therefore, it only needs to be sent once, although there is no harm in including it with each configuration.

Errors: An INVALID PARAMETER error will be returned if the parameter is not 'ON' or 'OFF'.

Example: BOOT=ON; // Turn on the PC5 supplies at power-up

5.1.8 CON1 – Select signal for AUX1 Connector

[PX5 & DP5G]

Command: CON1
Parameter: [DAC|AUXOUT1|AUXIN1]
Units: none
Default: DAC (PX5); AUXIN1 (DP5G)

Dependencies: none
Supported: FW6.03 and later; PX5 and DP5G [DP5 uses DACO, AUO1 & AUO2 to select signals. PX5 & DP5G use CONx commands to route selected signals to external connectors.]
Related: CON2, DACO, AUO1, AUO2, GPIN

Description: Selects the signal for the AUX1 connector on PX5 or DP5G/PCG. 'DAC' routs the output DAC to the AUX1 connector. ('DACO' selects which signal the DAC will output.) 'AUXOUT1' will output the digital signal selected by the 'AUO1' command. 'AUXIN1' switches the AUX1 connector to an input – 'AUXIN1' can then be selected as an input to the General Purpose Counter (via GPIN).

Errors: An INVALID PARAMETER error will be returned if the parameter is not one of the options specified.

Example: CON1=AUXOUT1; // Configure Connector 1 to AUXOUT1

5.1.9 CON2 – Select signal for AUX2 Connector

[PX5 & DP5G]

Command: CON2
Parameter: [AUXOUT2|AUXIN2|GATEH|GATEL]
Units: none
Default: AUXOUT2 (PX5); AUXIN2 (DP5G)

Dependencies: none
Supported: FW6.03 and later; PX5 and DP5G [DP5 uses DACO, AUO1, AUO2 & GATE to select signals. PX5 & DP5G use CONx commands to route selected signals to external connectors.]
Related: CON1, AUO1, AUO2, GPIN, GATE

Description: Selects the signal for the AUX2 connector on PX5 or DP5G/PCG. 'AUXOUT2' will output the digital signal selected by the 'AUO2' command. 'AUXIN2' switches the AUX1 connector to an input – 'AUXIN2' can then be selected as an input to the General Purpose Counter (via GPIN). 'GATEH' or 'GATEL' switches the AUX2 connector to an input, and uses the AUX2 signal as an active-high or active-low gate input.

Errors: An INVALID PARAMETER error will be returned if the parameter is not one of the options specified.

Example: CON2=AUXOUT2; // Configure Connector 2 to AUXOUT2

5.1.10 CLCK - Select FPGA Clock

Command: CLCK
Parameter: [20|80|AU{TO}] [DP5/PX5]
[20] [DP5G]
Units: MHZ
Default: AUTO [DP5/PX5]
20 [DP5G]

Dependencies: none
Supported: FW6.00 and later
Related: TPEA

Description: CLCK sets the FPGA clock to either 20MHz or 80MHz. If 'AUTO' is used as the parameter, then the value of the peaking time command (TPEA) will determine whether the FPGA clock is 20MHz or 80MHz. [The DP5G only supports 20MHz operation].

Errors: An INVALID PARAMETER error will be returned if the parameter is not one of those listed.

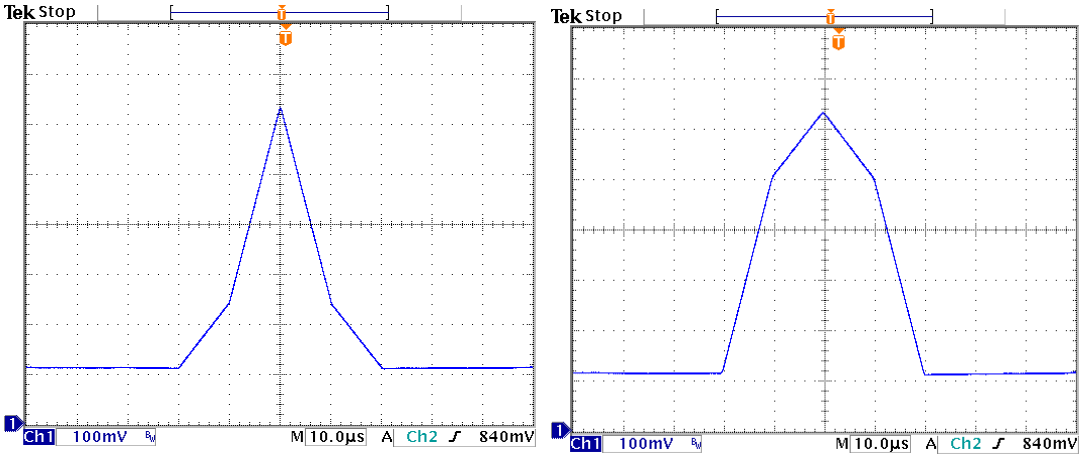
Example: CLCK=20; // This sets the FPGA clock to 20MHz

5.1.11 CUSP - Specify Non-Trapezoidal Shaping

Command: CUSP
Parameter: [{+|-}##|OF{F}]
Units: %
Default: OFF (same as 0%)
Range: -99 to +99; ~3.1% precision

Dependencies: none
Supported: FW6.00 and later
Related: TPEA

Description: CUSP determines whether the shaper produces a trapezoidal pulse shape (CUSP=0% or OFF); a cusp-like shape (parameter is positive); or a Gaussian-like shape (parameter is negative.) Below are scope captures, with Peaking Time set to 20uS and Flat Top set to 0uS; on the left is CUSP=+50%. On the right is CUSP=-50%.



Errors: An INVALID PARAMETER error will be returned if the parameter is a number outside the specified range. A text parameter is the same as OFF and will not return an error.

Example: CUSP=50; // Enable cusp-like shaping with 50% depth

5.1.12 DACF - Set DAC Offset

Command: DACF
Parameter: [{+|-}###]
Units: MV (millivolts)
Range: -500 to +499 mV (~3.9mV precision)
Default: 0mV

Dependencies: none
Supported: FW6.00 and later
Related: DACO, TPMO

Description: DACF sets the offset for the output DAC. The DAC has on output range of 0-1V – changing the offset allows one to see signals that would otherwise be out of range of the DAC. With a setting of 0, the DP5 clamps signals below 0V to 0V, and above 1V to 1V. At other DACF settings, the signal wraps around. [Note: if the test pulser function is enabled, then this command sets the amplitude of the test pulse. See TPMO for details.]

Errors: An INVALID PARAMETER error will be returned (and the DAC offset set to 0) if the commanded threshold is outside the range given.

Example:
```
DACF=30;    // This sets the DAC offset to 30mV, to allow signals
            // slightly below logical ground to be seen.
```

5.1.13 DACO - Select Signal for Output DAC

Command: DACO
Parameter: [#|OFF|FAST|SHAPED|INPUT|PEAK]
Units: none
Default: OFF
Range: 1-8, if a number is specified

Dependencies: none
Supported: FW6.00 and later
Related: DACF, CON1 (PX5/DP5G)

Description: DACO selects the signal to be output via the DAC. Generally, this is used for diagnostic purposes – the various signals can be inspected on an oscilloscope. Either the signal name can be used, or an index number, starting with 1 = 'FAST'. The selected signal is also fed into the Digital Oscilloscope. (See SCOE, SCOT and SCOG). Note: if the Test Pulser function is enabled, the DAC will output the test pulse signal while the Digital Oscilloscope will continue to capture the selected signal. **Note: PX5 and DP5G require the 'CON1' command to connect the output DAC to the AUX1 connector.**

Errors: An INVALID PARAMETER error will be returned if the parameter is not 1-8, or the signal name exactly as listed. [Most of the commands which have text parameters only require the first two characters; this command requires all characters specified.]

Example:
```
DACO=SHAPED;        // Select the shaped channel for the
                    // output DAC
```

5.1.14 GAIA - Set the Analog Gain Index

Command: GAIA
Parameter: [##]
Units: none
Default: none
Range: 1 to 16 [DP5]
1 to 28 [PX5]
1 to 4 [DP5G]

Dependencies: none
Supported: FW6.00 and later
Related: GAIF, GAIN

Description: GAIA selects the analog gain. The parameter specified is an index; it selects the analog gain from the list below. The first one listed has an index of 1, up to the last one, which has an index of 16 (DP5), 28 (PX5), or 4 (DP5G).

Available DP5 analog gains: 1.12, 2.49, 3.78, 5.28, 6.56, 8.39, 10.09, 11.32, 14.55, 17.77, 22.40, 30.84, 38.12, 47.47, 66.16, 101.83

Available PX5 analog gains: 1.00, 1.25, 1.56, 1.95, 2.43, 3.04, 3.78, 4.75, 5.96, 7.47, 9.34, 11.66, 14.57, 18.18, 22.72, 28.37, 35.46, 44.64, 55.83, 69.79, 87.10, 108.62, 135.84, 169.80, 211.99, 264.99, 330.70, 413.38

Available DP5G analog gains: 1.57, 2.56, 4.09, 6.77

Errors: An INVALID PARAMETER error will be returned if the parameter is not with the *range* specified.

Example: GAIA=5; // Select an analog gain of 6.56 (DP5)

5.1.15 GAIF - Set the Fine Gain

Command: GAIF
Parameter: [##.####]
Units: none
Default: none
Range: 0.5-1.9999

Dependencies: TPEA must precede GAIF
Supported: FW6.00 and later
Related: GAIA, GAIN

Description: GAIF sets the fine gain. The total gain is a product of the analog gain and fine gain. [Generally, either the GAIN command is used, or the combination of GAIA and GAIF.]

Internally, the fine gain is combined with a normalization for gain variations due to peaking time. Because of this, the precision of the fine gain is dependent on peaking time. It is always has a precision of at least 1 channel in 8192; generally, the precision is better than that.

Errors: An INVALID PARAMETER error will be returned if the parameter is not with the *range* specified.

Example: GAIF=1.05; // Set the digital fine gain to 1.05

5.1.16 GAIN - Set the Total Gain

Command: GAIN
Parameter: [###.###]
Units: none
Default: none
Range: 0.75 to 150 [DP5]
0.75 to 500 [PX5]
1.0 to 10.0 [DP5G]

Dependencies: TPEA must precede GAIN
Supported: FW6.00 and later
Related: GAIA, GAIF

Description: GAIN sets the total gain. First, it searches for the nearest available 'coarse gain' for the analog front end (listed below for reference.) It then calculates the digital 'fine gain' to achieve the commanded gain, while also normalizing for gain differences caused by peaking time variations.

(See the GAIA command for available analog gains)

Errors: An INVALID PARAMETER error will be returned if the parameter is not with the *range* specified.

Example: GAIN=100; // Set the combined digital and analog gain to 100

5.1.17 GATE - Configure the GATE Input

[DP5 only]

Command: GATE
Parameter: [OF{F}|HI{GH}|LO{W}]
Units: none
Default: OFF

Dependencies: none
Supported: FW6.00 and later; DP5 only [PX5 and DP5G use 'CON2' to configure the GATE function]
Related: CON1 (PX5 & DP5G)

Description: GATE selects whether the GATE function is enabled, and if so, the polarity of the GATE input. "HIGH" means that events are rejected while the GATE input is high; "LOW" means that events are rejected while the GATE input is low. **If enabled, GATE uses the DP5 AUX_IN1 input, which has a pull-down resistor on it**.

Errors: An INVALID PARAMETER error will be returned if the parameter is not one of the values specified.

Example:
```
GATE=HIGH;      // Enable GATE – events are rejected while input
                // is high
```

5.1.18 GPED - Select General Purpose Counter Edge

Command: GPED
Parameter: [RI{SING}|FA{LLING}]
Units: none
Default: FALLING

Dependencies: none
Supported: FW6.00 and later
Related: GPIN, GPME, GPGA, GPMC; 'Clear G.P. Counter' Request Packet

Description: GPED selects on which edge of its source the General Purpose (G.P.) Counter will increment.

Errors: An INVALID PARAMETER error will be returned if the parameter is not one of the values specified.

Example: GPED=RI; // Count on the rising edge of the G.P. Counter input

5.1.19 GPGA - General Purpose Counter Uses GATE

Command: GPGA
Parameter: [ON|OF{F}]
Units: none
Default: ON

Dependencies: none
Supported: FW6.00 and later
Related: GPED, GPIN, GPME, GPMC, GATE; 'Clear G.P. Counter' Request Packet

Description: GPGA selects whether the G.P. counter is conditioned by the GATE input. In order to use this, the GATE command must also be sent to configure the GATE input polarity. If GATE is disabled ('GATE=OFF'), then GPGA will have no effect. GPGA is redundant if GPME=ON, because MCA Enable is gated by GATE already.

Errors: An INVALID PARAMETER error will be returned if the parameter is not one of the values specified.

Example: GPGA=ON; // Qualify the G.P. Counter with GATE

5.1.20 GPIN - Select the Source for the General Purpose Counter

Command: GPIN
Parameter: [#|AUX1|AUX2|PILEUP|RTDREJ|SCA8|TBD|DETRES|OFF]
Units: none
Default: AUX1
Range: 1-8

Dependencies: none
Supported: FW6.00 and later
Related: GPED, GPME, GPGA, GPMC; 'Clear G.P. Counter' Request Packet; CON1, CON2 (PX5/DP5G)

Description: GPIN selects the source for the General Purpose (G.P.) Counter. The G.P. Counter will count rising or falling edges of the source (see GPED). ('AUX1' refers to the DP5 AUX_IN1 input; 'AUX2 ' refers to the DP5 AUX_IN2 input. 'TBD' is currently undefined.) An index can be used rather than the signal name; 1=AUX1…8=OFF. Note: PX5 and DP5G require the 'CON1' ('CON2') command to connect the AUXIN1 (AUXIN2) input to the AUX1 (AUX2) connector, for it to be available as a counter source.

Errors: An INVALID PARAMETER error will be returned if the parameter is not one of the values specified, or a number in the range specified, in which case the default of AUX1 will be used. [Most of the commands which have text parameters only require the first two characters; this command requires all characters specified.]

Example:
```
GPIN=RTDREJ;        // Set the G.P. Counter to count events
                    // rejected by RTD
```

5.1.21 GPMC - General Purpose Counter is Cleared with MCA

Command: GPMC
Parameter: [ON|OF{F}]
Units: none
Default: ON

Dependencies: none
Supported: FW6.00 and later
Related: GPED, GPIN, GPME, GPGA, GATE; 'Clear G.P. Counter' Request Packet

Description: GPMC selects whether the G.P. counter is cleared when the MCA (spectrum) is cleared. If GPMC=OFF, then the G.P. Counter is only cleared by the 'Clear G.P. Counter' Request Packet.

Errors: An INVALID PARAMETER error will be returned if the parameter is not one of the values specified.

Example:
```
GPMC=ON;        // Clear the G.P. Counter when the spectrum is
                // cleared
```

5.1.22 GPME - General Purpose Counter Uses MCA Enable

Command: GPME
Parameter: [ON|OF{F}]
Units: none
Default: ON

Dependencies: none
Supported: FW6.00 and later
Related: GPED, GPIN, GPGA, GPMC; 'Clear G.P. Counter' Request Packet

Description: GPME selects whether the G.P. Counter is gated by the internal MCA Enable signal, or whether it is free-running. [The MCA is disabled by Detector Reset (if enabled – see RESL), GATE (if enabled – see GATE), Preset Counts (see PREC), Preset Time (see PRET), Preset Real Time (see PRER), the 'MCA Enable' and 'MCA Disable' Request Packets, and briefly during internal buffer operations.]

Errors: An INVALID PARAMETER error will be returned if the parameter is not one of the values specified.

Example: GPME=ON; // Enable/disable the G.P. Counter with the MCA

5.1.23 HVSE - Turn On/Off the PC5 High Voltage Supply

Command: HVSE
Parameter: [{+|-}####|OFF]
Units: V
Default: OFF
Range: 0 to 1499V (for a positive HV PC5);
0 to -1499V (for a negative HV PC5);
-1499V to 1499V [PX5]
~0.4V precision

Dependencies: none
Supported: FW6.00 and later

Description: HVSE turns on or off the PC5 HV supply and sets the HV. The PC5 can have either a positive or negative supplies. The DP5 will compare the polarity of the parameter to the polarity of the PC5 HV supply, and will only turn on the HV supply if the specified polarity matches the PC5. [PX5 compares the polarity of the parameter to the position of the HV polarity jumper.]

Errors: DP5: An INVALID PARAMETER error will be returned if the PC5 does not have the same polarity HV supply as the voltage specified, or if the specified HV is outside the range listed. A PC5 NOT PRESENT error will be returned if a PC5 is not detected.
PX5: An INVALID PARAMETER error will be returned if the polarity of the specified voltage does not agree with the PX5 HV polarity jumper, or if the specified HV is outside the range listed.

Example:
```
HVSE=150;          // Set the PC5 HV supply to 150V, but only
                   // if it's a +HV PC5
```

5.1.24 INOF - Set the Input Offset

[DP5/PX5 only]

Command: INOF
Parameter: [{+|-}####|AU{TO}|DE{F}]
Units: MV (millivolts)
Default: DEF
Range: -2047mV to +2047mV

Dependencies: AINP, if 'DEF' is used; INOG, if INOG=HIGH (see below)
Supported: FW6.00 and later; not supported on DP5G, which does not have a variable input offset
Related: AINP, INOG

Description: INOF sets a DAC, which shifts the input to the ADC so that the signal is in the proper range. Generally, the default setting ('DEF') can be used, because a standard DP5 has an AC-coupled signal input which removes DC levels from the preamp. For a customized DP5 (or a PX5 with preamp pole cancellation enabled via 'PAPZ'), it may be necessary to set the input offset manually. If so, it should be set to produce a DC level of ~200mV (for AINP=NEG) or ~1.8V (for AINP=POS) at the test point AMP3OUT (DP5) or AMPOUT (PX5). [Note that in the DP5, there is a gain of 2 for the DAC output; changing INOF by 100mV results in the AMP3OUT testpoint shifting 200mV.]

In the PX5, the 'INOG' command can be used to change the gain of the input offset. If used, the 'INOG' command must precede INOF. If the high gain input offset ('INOG=HIGH') is used, then the 'DEF' (default) option for INOF cannot be used, and will result in an INVALID PARAMETER error.

The 'AUTO' parameter may be specified, in which case the DP5 or PX5 will search for an appropriate input offset. Note that the detector must be stable for this to work; if HV or temperature has not stabilized, then this will not operate properly.

Errors: An INVALID PARAMETER error will be returned if the parameter is not one of those listed.

Example: INOF=200; // Set the input offset DAC to 200mV

5.1.25 INOG - Set the Input Offset Gain

[PX5 only]

Command: INOG
Parameter: [LO{W}|HI{GH}]
Units: none
Default: LOW

Dependencies:
Supported: FW6.06 and later; PX5 only
Related: AINP, INOF

Description: In the PX5, the input offset is normally injected after the gain stages – this is the "LOW" setting. If preamp pole cancellation is used, then a DC offset may exist prior to the gain stages, and depending on gain, may result in an offset too large for the low gain input offset to correct. Selecting "HIGH" results in the input offset being injected earlier in the analog chain, so that it can correct a larger DC offset.

Errors: An INVALID PARAMETER error will be returned if the parameter is not one of those listed.

Example: INOG=HIGH; // Set the input offset gain to HIGH

5.1.26 MCAC - Select Number of MCA Channels

Command: MCAC
Parameter: [256|512|1024|2048|4096|8192]
Units: channels
Default: 1024

Dependencies: none
Supported: FW6.00 and later

Description: MCAC selects the total number of channels for the MCA. [This is sometimes referred to as 'MCA Gain'.]

Errors: An INVALID PARAMETER error will be returned if the parameter is other than those listed, and 1024 channel operation will be selected.

Example: MCAC=512; // Select 512 MCA channels

5.1.27 MCAE - Initial State of MCA Enable

Command: MCAE
Parameter: [ON|OF{F}]
Units: none
Default: OFF

Dependencies: none
Supported: FW6.00 and later

Description: MCAE sets the initial state of MCA Enable. It determines whether the acquisition starts immediately after the Configuration Packet is processed ('ON'), or whether the DP5 will wait for an 'Enable MCA/MCS' Request Packet to start the acquisition ('OFF').

Errors: An INVALID PARAMETER error will be returned if the parameter is neither 'ON' nor 'OFF'.

Example: MCAE=ON; // Start taking data immediately

5.1.28 MCAS - Select the MCA Source

Command: MCAS
Parameter: [NO{RM}|MC{S}|FA{ST}| PU{R}|RT{D}]
Units: none
Default: NORM

Dependencies: RTDE
Supported: FW6.00 and later
Related:

Description: MCAS selects the operating mode of the MCA, as listed below:
'NORM' – normal MCA operation, using the shaped channel as the source

'MCS' - selects Multi-channel Scaler mode, rather than MCA. [MCS mode records counts vs. time, rather than counts vs. energy. SCA8 is the source of counts in MCS mode. See SCAI, SCAL, SCAH]

'FAST' – the MCA produces a spectrum using the Fast channel as the source, rather than the shaped channel. [This requires RTD to be disabled to work properly. See RTDE.]

'PUR' – the MCA records only piled-up events, rather than normal events. [This requires RTD to be disabled to work properly. See RTDE.]

'RTD' – the MCA records only events that would normally be rejected by RTD. [This requires RTD to be enabled to work properly. See RTDE, RTDS and RTDT.]

Errors: An INVALID PARAMETER error will be returned if the parameter is not one of those listed, and 'NORM' will be used.

Example: MCAC=FAST; // Acquire a spectrum of the fast channel

5.1.29 MCSL - Set Low Threshold for MCS

Command: MCSL
Parameter: [####]
Units: channel
Default: 0
Range: Depends on number of channels specified by MCAC:
256 ch: 0 - 255
512 ch: 0 – 511
1024 ch: 0 - 1023
2048 ch: 0 - 2047
4096 ch: 0 - 4095
8192 ch: 0 – 8191

Dependencies: none
Supported: FW6.01 and later
Related: MCSH, MCST, MCAS

Description: MCSL sets the low threshold for the MCS (Multi-Channel Scaler). The low and high thresholds are exclusive; for example, PRCL=100 and PRCH=102 will only count events that occur in channel 101. [The MCS thresholds only have meaning when the MCS operational mode is selected by the MCAS command.]

Errors: An INVALID PARAMETER error will be returned if the parameter is outside the range 0- 8191; it is not error-checked with regard to the ranges listed. If MCSL is set higher than MCSH, no error is generated, but no events will be counted as this is an impossible condition.

Example:
```
MCSL=100;        // Set MCS to count events from
MCSH=200;        //  channel 101 to 199
```

5.1.30 MCSH - Set High Threshold for MCS

Command: MCSH
Parameter: [####]
Units: channel
Default: 8191
Range: Depends on number of channels specified by MCAC:
256 ch: 0 - 255
512 ch: 0 – 511
1024 ch: 0 - 1023
2048 ch: 0 - 2047
4096 ch: 0 - 4095
8192 ch: 0 – 8191

Dependencies: none
Supported: FW6.01 and later
Related: MCSL, MCST, MCAS

Description: MCSH sets the high threshold for the MCS (Multi-Channel Scaler). The low and high thresholds are exclusive; for example, MCSL=100 and MCSH=102 will only count events that occur in channel 101. [The MCS thresholds only have meaning when the MCS operational mode is selected by the MCAS command.]

Errors: An INVALID PARAMETER error will be returned if the parameter is outside the range 0- 8191; it is not error-checked with regard to the ranges listed. If MCSL is set higher than MCSH, no error is generated, but no events will be output or counted as this is an impossible condition.

Example:
```
MCSL=100;        // Set MCS to count events from
MCSH=200;        //  channel 101 to 199
```

5.1.31 MCST - Set the MCS Timebase

Command: MCST
Parameter: [###.###]
Units: S (seconds)
Default: 0
Range: 0 – 655.35s; 10mS precision

Dependencies: none
Supported: FW6.00 and later
Related: MCAS

Description: MCST configures the timebase used by the MCS (Multi-Channel Scaler) – this is the duration of each 'channel' in the MCS acquisition. It is only relevant if the acquisition is in MCS mode, rather than MCA mode. (See the MCAS command.)

Errors: An INVALID PARAMETER error will be returned if the parameter is not within the range specified.

Example: MCST=10; // Set the MCS timebase to 10 seconds per channel

5.1.32 PAPS - Turn On/Off the Preamp Power Supplies

[DP5/PX5 only]

Command: PAPS
Parameter: [8{.5}|5|OF{F}|ON] [DP5]
[8{.5}|5|OF{F}] [PX5]
Units: V
Default: OFF

Dependencies: none
Supported: FW6.00 and later; not supported on DP5G

Description: PAPS turns on or off the PC5 preamp power supplies. The PC5 can have either +/-5V or +/-8.5V preamp supplies. If the 5 or 8.5 option is specified, the PC5 will be checked to verify if it's the commanded type before it is switched on. If 'ON' is commanded, this test will not be performed – the PC5 preamp supplies will be turned on regardless of which voltage they are. ['ON' is not an option for the PX5 – the voltage must be specified.]

Errors: An INVALID PARAMETER error will be returned if the PC5 does not have the preamp power supplies of the voltage specified, or if a parameter other than those listed is specified. A PC5 NOT PRESENT error will be returned if a PC5 is not detected.

Example:
```
PAPS=5;          // Turn on the PC5 preamp supplies, but only
                 // if it's a 5V PC5
```

5.1.33 PAPZ – Preamp Pole-Zero Cancellation

[PX5 only]

Command: PAPZ
Parameter: [####.#|OF{F}]
Units: US (microseconds)
Default: OFF
Range: 34.5-4387uS [firmware rounds to nearest of 127 discrete values; discrete values are 4386.7uS/n, for n=1…127]

Dependencies: none
Supported: FW6.05 and later; PX5 only

Description: PAPZ enables or disables the pole-zero cancellation in the PX5 front-end electronics, and selects the time constant to cancel. The specified time constant is rounded to the nearest discrete value. The equation to generate the discrete values is give in the 'Range' above. These values have a tolerance of roughly 1%. The 127 possible values, in uS, are:

34.5	39.5	46.2	55.5	69.6	93.3	141.5	292.4
34.8	39.9	46.7	56.2	70.8	95.4	146.2	313.3
35.1	40.2	47.2	57.0	71.9	97.5	151.3	337.4
35.4	40.6	47.7	57.7	73.1	99.7	156.7	365.6
35.7	41.0	48.2	58.5	74.4	102.0	162.5	398.8
36.0	41.4	48.7	59.3	75.6	104.4	168.7	438.7
36.3	41.8	49.3	60.1	77.0	107.0	175.5	487.4
36.6	42.2	49.8	60.9	78.3	109.7	182.8	548.3
36.9	42.6	50.4	61.8	79.8	112.5	190.7	626.7
37.2	43.0	51.0	62.7	81.2	115.4	199.4	731.1
37.5	43.4	51.6	63.6	82.8	118.6	208.9	877.3
37.8	43.9	52.2	64.5	84.4	121.9	219.3	1096.7
38.1	44.3	52.9	65.5	86.0	125.3	230.9	1462.2
38.5	44.8	53.5	66.5	87.7	129.0	243.7	2193.4
38.8	45.2	54.2	67.5	89.5	132.9	258.0	4386.7
39.2	45.7	54.8	68.5	91.4	137.1	274.2	

Errors: Parameter values between 34.5uS and 8773uS are rounded to the nearest discrete value from the table above. Outside that range, an INVALID PARAMETER error will be returned, and the pole-zero cancellation circuit will be switched off.

Example:
```
PAPZ=50;        // Turn on the pole-zero cancellation, and
                //  cancel a 50uS time constant
```

5.1.34 PDMD - Select the Peak Detect Mode

Command: PDMD
Parameter: [NO{RM}|MI{N}]
Units: none
Default: NORM

Dependencies: none
Supported: FW6.00 and later
Related: SOFF

Description: Normally, the DP5 Peak Detect unit searches for maxima in the shaped pulse stream, to determine which events to include in the spectrum. (This is the normal Peak Detect Mode; PDMD=NORM). PDMD=MIN configures the Peak Detect unit to search for both minima and maxima, in order for the MCA to be able to capture the 'noise Gaussian' around the zero energy point of the spectrum. Because the MCA can't capture negative peaks, the Spectrum Offset (SOFF) should be used to shift the spectrum positive, so the entire noise Gaussian can be captured.

Errors: An INVALID PARAMETER error will be returned if the parameter is not NORM or MIN.

Example: PDMD=MIN; // Set the Peak Detect to MIN+MAX mode

5.1.35 PRCL - Set Low Threshold for Preset Counts

Command: PRCL
Parameter: [####]
Units: channel
Default: 0
Range: Depends on number of channels specified by MCAC:
256 ch: 0 - 255
512 ch: 0 – 511
1024 ch: 0 - 1023
2048 ch: 0 - 2047
4096 ch: 0 - 4095
8192 ch: 0 – 8191

Dependencies: none
Supported: FW6.01 and later
Related: PRCH, PREC

Description: PRCL sets the low threshold for the Preset Counts. The low and high thresholds are exclusive; for example, PRCL=100 and PRCH=102 will only count events that occur in channel 101.

Errors: An INVALID PARAMETER error will be returned if the parameter is outside the range 0- 8191; it is not error-checked with regard to the ranges listed. If PRCL is set higher than PRCH, no error is generated, but no events will be counted as this is an impossible condition.

Example:
```
PRCL=100;        // Set Preset Counts to count events from
PRCH=200;        //  channel 101 to 199
```

5.1.36 PRCH - Set High Threshold for Preset Counts

Command: PRCH
Parameter: [####]
Units: channel
Default: 8191
Range: Depends on number of channels specified by MCAC:
256 ch: 0 - 255
512 ch: 0 – 511
1024 ch: 0 - 1023
2048 ch: 0 - 2047
4096 ch: 0 - 4095
8192 ch: 0 – 8191

Dependencies: none
Supported: FW6.01 and later
Related: PRCL, PREC

Description: PRCH sets the high threshold for the Preset Counts. The low and high thresholds are exclusive; for example, PRCL=100 and PRCH=102 will only count events that occur in channel 101.

Errors: An INVALID PARAMETER error will be returned if the parameter is outside the range 0- 8191; it is not error-checked with regard to the ranges listed. If PRCL is set higher than PRCH, no error is generated, but no events will be output or counted as this is an impossible condition.

Example:
```
PRCL=100;       // Set Preset Counts to count events from
PRCH=200;       //  channel 101 to 199
```

5.1.37 PREC - Preset Counts

Command: PREC
Parameter: [##########|OF{F}]
Units: counts
Default: OFF
Range: 0 – 4,294,967,295 (i.e. 2^32 -1)

Dependencies: none
Supported: FW6.00 and later
Related: PRET, PRER, PRCL, PRCH

Description: PREC specifies the preset counts. The acquisition will stop automatically when the number of events registered between the channels specified by PRCL and PRCH reaches this value. If OFF (or 0) is specified, the acquisition will continue unless stopped by other means (PRET, PRER, MCA disable Request Packet, etc.) By configuring PREL and PREH appropriately, this command allows an acquisition to be stopped by the counts in a single channel, a ROI, or the entire spectrum.

Errors: No errors are returned – an unrecognized parameter is interpreted as 'OFF'. A number outside the range listed is converted to a 32-bit unsigned integer.

Example:
```
PREC=10000;     // Stop when the total counts between channels PRCL & PRCH
                //  reaches 10,000
```

5.1.38 PRER - Preset Real Time

Command: PRER
Parameter: [########.###|OF{F}]
Units: S (seconds)
Default: OFF
Range: 0 – 4,294,967.29 seconds; 0.01 second precision
(Note that because the parameter field is limited to 10 characters, the following limit applies if 3 digits are required to the right of the decimal point)
0 – 999,999.999 seconds; 0.001 second precision

Dependencies: none
Supported: FW6.01 and later
Related: PREC, PRET

Description: PRER sets the preset real time. The acquisition will stop automatically when the real time reaches this value. If OFF (or 0) is specified, the acquisition will continue unless stopped by other means (PREC, PRET, MCA disable Request Packet, etc.)

Errors: An INVALID PARAMETER error will be returned if the parameter is not 'OFF' or a number within the range specified.

Example: PRER=100; // Stop the acquisition when real time reaches 100s

5.1.39 PRET - Preset Acquisition Time

Command: PRET
Parameter: [########.#|OF{F}]
Units: S (seconds)
Default: OFF
Range: 0 – 99,999,999.9 seconds; 0.1 second precision

Dependencies: none
Supported: FW6.00 and later
Related: PREC, PRER

Description: PRET sets the preset acquisition time. The acquisition will stop automatically when the acquisition time reaches this value. If OFF (or 0) is specified, the acquisition will continue unless stopped by other means (PREC, PRER, MCA disable Request Packet, etc.)

Errors: An INVALID PARAMETER error will be returned if the parameter is not 'OFF' or a number within the range specified.

Example: PRET=100; // Stop the acquisition when it reaches 100s

5.1.40 PURE - Pile-up Reject Enable

Command: PURE
Parameter: [ON|OF{F}|MA{X}|###.###]
Units: US (microseconds)
Default: OFF
Range: The table below specifies the range of the PUR interval as a function of peaking time and FPGA clock.

Dependencies: TPEA must precede PURE, if ###.### is specified. None, otherwise.
Supported: FW6.00 and later
Related: THFA

Description: PURE is used to enable or disable Pile-up Rejection. Normally (for PURE=ON), the Pile-up interval is dynamic; the DP5 varies the interval on a pulse-by-pulse basis to minimize deadtime. However, this can be overridden. The interval can be set to a constant, or to the maximum available. See the table below for allowable values. The PUR interval is started when a peak is detected on the Fast Channel which exceeds its threshold, so THFA must be set properly.

Errors: An INVALID PARAMETER error will be returned if the parameter is not ON, OFF, MAX, or within the range specified below. If a number is specified that is greater than is allowed, INVALID PARAMETER will be returned and the PUR interval will be set to its maximum.

Example: PURE=ON; // Enable Pile-up Rejection

FPGA Clock=20MHz			FPGA Clock=80MHz		
Peaking Time	**PUR Interval**	**Precision**	**Peaking Time**	**PUR Interval**	**Precision**
0.800 - 6.40uS	0 - 25.6uS	50nS	0.200 - 1.600uS	0 - 6.39uS	12.5nS
6.60 - 12.80uS	0 - 51.1uS	100nS	1.650 - 3.200uS	0 - 12.8uS	25nS
13.20 - 25.60uS	0 - 102.2uS	200nS	3.30 - 6.40uS	0 - 25.6uS	50nS
26.40 - 51.20uS	0 - 204.4uS	400nS	6.60 - 12.80uS	0 - 51.1uS	100nS
52.80 - 102.4uS	0 - 408.8uS	800nS	13.20 - 25.60uS	0 - 102.2uS	200nS

5.1.41 RESC - Reset the Configuration to Defaults

Command: RESC
Parameter: [Y]
Units: none
Default: none

Dependencies: none
Supported: FW6.00 and later
Related:

Description: RESC is used to reset the DP5's configuration to default settings. It resets each setting to the default listed for each command and in table XX. If it is used, it should be the first command in the ACII configuration packet.

RESC is useful because the DP5 stores the settings from previous configuration commands in non-volatile memory, and new

Errors: No error is returned regardless of the parameter, but the configuration is only reset if the parameter is 'Y'.

Example: RESC=Y; // Reset the configuration to default settings

5.1.42 RESL - Select Reset Lockout Interval

Command: RESL
Parameter: [#####|OF{F}]
Units: US (microseconds)
Range: In FW6.01, the DP5 will select the nearest available Reset Lockout Interval from the following list:

FPGA clock=80MHz: 25.6uS, 51.2uS, 102.4uS, 204.8uS, 409.6uS, 819.2uS, 1638.4uS, 3276.8uS

FPGA clock=20MHz: 102.4uS, 204.8uS, 409.6uS, 819.2uS, 1638.4uS, 3276.8uS, 6553.6uS, 13107.2uS

Default: OFF

Dependencies: TPEA must precede RESL
Supported: FW6.00 and later
Related:

Description: RESL determines whether the MCA data taking will be paused when a detector reset is detected. If OFF is selected, then the detection of detector reset is disabled. Otherwise, the nearest value from those listed in *Range* will be selected.

Errors: No error checking is done on the parameter; the next largest value in *Range* will be used if a number is specified. If the parameter isn't recognized, the lowest value in *Range* is used.

Example: RESL=1000; // This selects a Reset Lockout Interval of 1638.4uS

5.1.43 RTDD - Set Custom RTD Oneshot Delay

Command: RTDD
Parameter: [###]
Units: decimated clocks
Default: none
Range: 1 - 127

Dependencies: RTDE must precede RTDD
Supported: FW6.00 and later
Related: RTDE, RTDS, RTDT, RTDW

Description: RTDD is used to fine-tune the timing of the RTD logic. It may be needed for events with very slow risetimes, or configurations where the shaper produces a non-trapezoidal/cusp shape. For these events, the peak of the shaped event may not occur when it is expected. The standard RTD timing parameters are calculated when the RTDE command is received, so the RTDW & RTDD commands must come after RTDE, so that they can override the standard timing.

To fine-tune the RTD timing:

1. Use DACO=SHAPED and AUO2=5 (for RTD_ONESHOT). Use an oscilloscope to view both signals. Trigger on either.

2. Use the parameters for RTDW and RTDD to adjust the width and delay of the RTD_ONESHOT signal – it should bracket the time during which the peak occurs for a variety of pulses.

Errors: An INVALID PARAMETER error will be returned if the parameter is not in the *range* specified.

Example:
```
RTDD=60;        // Override the standard RTD timing – set the
                // RTD Oneshot delay to 60 decimated clocks
```

5.1.44 RTDE - Enable Risetime Discrimination

Command: RTDE
Parameter: [ON|OF{F}]
Units: none
Default: OFF

Dependencies: TPEA & TFLA
Supported: FW6.00 and later
Related: RTDT, RTDS

Description: RTDE is used to enable risetime discrimination (RTD). RTDS and RTDT are also needed to completely configure RTD. For very slow or distorted pulses, RTDW and RTDD may be needed, to override standard RTD timing parameters.

Errors: An INVALID PARAMETER error will be returned if the parameter is not ON or OFF, and RTD will be disabled.

Example: RTDE=ON; // Enable Pile-up Rejection

5.1.45 RTDS - Set Risetime Discrimination Sensitivity

Command: RTDS
Parameter: [####]
Units: %
Default: 0%
Range: 2-398% (1.5% precision), 400-1593% (6.5% precision)

Dependencies: none
Supported: FW6.00 and later
Related: RTDE, RTDT

Description: RTDS sets the Risetime Discrimination (RTD) Sensitivity. RTD is performed by taking the ratio of the peak slow channel amplitude to the peak fast channel amplitude for a given pulse – the event is rejected if the ratio is above the sensitivity setting. [As the risetime gets slower, the fast channel response falls off faster than the slow channel, so the ratio increases for the event increases.] If the peak slow channel amplitude is below the RTD Threshold (see RTDT), it will be accepted regardless of the ratio and sensitivity setting. This command has no effect if RTD is disabled (see RTDE).

Errors: An INVALID PARAMETER error will be returned if the parameter is not within the range specified, and the sensitivity will be set to 0%.

Example: RTDS=500; // Set the RTD Sensitivity to 500%

5.1.46 RTDT - Set Risetime Discrimination Threshold

Command: RTDT
Parameter: [##.###]
Units: % full-scale
Default: 0%
Range: 0-49.9% (~0.2% precision)

Dependencies: none
Supported: FW6.00 and later
Related: RTDE, RTDS

Description: RTDT sets the Risetime Discrimination (RTD) Threshold. If the peak slow channel amplitude is below the RTD Threshold, it will be accepted regardless of the sensitivity setting. This has no effect if RTD is disabled (see RTDE).

Errors: An INVALID PARAMETER error will be returned if the parameter is not within the range specified, and the threshold will be set to 49.9% (or 0% if a negative number is specified).

Example:
```
RTDT=10;        // Set the RTD Threshold to 10% full-scale.
                // i.e. for 1K channels, 10% = channel 102
```

5.1.47 RTDW - Set Custom RTD Oneshot Width

Command: RTDW
Parameter: [###]
Units: decimated clocks
Default: none
Range: 1 - 127

Dependencies: RTDE must precede RTDW
Supported: FW6.00 and later
Related: RTDE, RTDS, RTDT, RTDD

Description: RTDW is used to fine-tune the timing of the RTD logic. It may be needed for events with very slow risetimes, or configurations where the shaper produces a non-trapezoidal/cusp shape. For these events, the peak of the shaped event may not occur when it is expected. The standard RTD timing parameters are calculated when the RTDE command is received, so the RTDW & RTDD commands must come after RTDE, so that they can override the standard timing.

For instructions on how to fine-tune the RTD timing, see the 'RTDD' command.

Errors: An INVALID PARAMETER error will be returned if the parameter is not in the *range* specified.

Example:
```
RTDW=60;        // Override the standard RTD timing – set the
                // RTD Oneshot period to 60 decimated clocks
```

5.1.48 SCAH - Set SCA High Threshold

Command: SCAH
Parameter: [####]
Units: channel
Default: 8191 (for all SCAs)
Range: Depends on number of channels specified by MCAC:
256 ch: 0 - 255
512 ch: 0 – 511
1024 ch: 0 - 1023
2048 ch: 0 - 2047
4096 ch: 0 - 4095
8192 ch: 0 – 8191

Dependencies: SCAI must precede SCAH
Supported: FW6.00 and later
Related: SCAI, SCAL, SCAO; SCAW is not indexed

Description: SCAH sets the high threshold for the SCA referenced by the SCA Index (SCAI). The low and high thresholds are exclusive; for example, SCAL=100 and SCAH=102 will only output events that occur in channel 101.

Errors: An INVALID PARAMETER error will be returned if the parameter is outside the range 0- 8191; it is not error-checked with regard to the ranges listed. If SCAL is set higher than SCAH, no error is generated, but no events will be output or counted as this is an impossible condition.

Example:
```
SCAI=2;          // Set the SCA Index to 2; enable SCA2 output as
                 // active high
SCAL=100;        // Set the SCA2 LLD to channel 100
SCAH=200;        // Set the SCA2 HLD to channel 200
```

5.1.49 SCAI - Set SCA Index

Command: SCAI
Parameter: [#]
Units: none
Default: none
Range: 1-16

Dependencies:
Supported: FW6.00 and later
Related: SCAL, SCAH, SCAO; SCAW is not indexed

Description: SCAI is used in conjunction with the SCAL, SCAH and SCAO commands to specify which SCA these commands will apply to. When the SCA index is set to 1 through 8, it also enables the specified SCA output and sets it to active high, so the SCAO command doesn't need to be specified unless a different output option is needed.

Errors: An INVALID PARAMETER error will be returned if the parameter is not within the range specified.

Example:
```
SCAI=2;          // Set the SCA Index to 2; enable SCA2 output as
                 // active high
SCAL=100;        // Set the SCA2 LLD to channel 100
SCAH=200;        // Set the SCA2 HLD to channel 200
```

5.1.50 SCAL - Set SCA Low Threshold

Command: SCAL
Parameter: [####]
Units: channel
Default: 0 (for all SCAs)
Range: Depends on number of channels specified by MCAC:
256 ch: 0 - 255
512 ch: 0 – 511
1024 ch: 0 - 1023
2048 ch: 0 - 2047
4096 ch: 0 - 4095
8192 ch: 0 – 8191

Dependencies: SCAI must precede SCAL
Supported: FW6.00 and later
Related: SCAI, SCAH, SCAO; SCAW is not indexed

Description: SCAL sets the low threshold for the SCA referenced by the SCA Index (SCAI). The low and high thresholds are exclusive; for example, SCAL=100 and SCAH=102 will only output events that occur in channel 101.

Errors: An INVALID PARAMETER error will be returned if the parameter is outside the range 0- 8191; it is not error-checked with regard to the ranges listed. If SCAL is set higher than SCAH, no error is generated, but no events will be output or counted as this is an impossible condition.

Example:
```
SCAI=2;          // Set the SCA Index to 2; enable SCA2 output as
                 // active high
SCAL=100;        // Set the SCA2 LLD to channel 100
SCAH=200;        // Set the SCA2 HLD to channel 200
```

5.1.51 SCAO - Select SCA Output Level

Command: SCAO
Parameter: [OF{F}|HI{GH}|LO{W}]
Units: none
Default: OFF; HIGH when SCAI is set for an SCA

Dependencies: SCAI must precede SCAO
Supported: FW6.00 and later
Related: SCAI, SCAL, SCAH; SCAW is not indexed

Description: SCAO configures the output of the indexed SCA. For a setting of 'OFF', the indexed SCA output will always be low. For a setting of 'HIGH', the indexed output will be normally low and will produce a high pulse when an event occurs between the SCA low and high thresholds. For a setting of 'LOW', the indexed output will be normally high and will produce a low pulse. Note: only SCAs 1-8 have pulse outputs; SCAs 9-16 have counters, not pulse outputs. Also, if the SCA outputs 1-8 are all OFF, then the outputs are all tri-stated. If any of the outputs 1-8 are enabled, then they all are driven.

Errors: An INVALID PARAMETER error will be returned if the parameter is not one of those specified in *range*. Selecting the output level when the SCA index is set to 9-16 has no meaning, since these SCAs don't have outputs, but this will not produce an error.

Example:
```
SCAI=2;          // Set the SCA Index to 2; enable SCA2 output as
                 // active high
SCAO=LOW;        // Set SCA2 to an active low output instead
```

5.1.52 SCAW - Select SCA Output Pulse Width

Command: SCAW
Parameter: [100|1000]
Units: NS (nanoseconds)
Default: 100

Dependencies: none
Supported: FW6.00 and later
Related: SCAI, SCAL, SCAH

Description: SCAW sets the output pulse width for SCA 1-8. (SCA 9-16 don't have outputs.) This setting isn't indexed like other SCA settings; it applies to all 8 SCAs.

Errors: An INVALID PARAMETER error will be returned if the parameter is not 100 or 1000, and the width will be set to 100.

Example: SCAW=1000; // Set SCA 1-8 to produce 1000nS pulses

5.1.53 SCOE - Set Digital Scope Trigger Edge

Command: SCOE
Parameter: [RI{SING}|FA{LLING}]
Units: none
Default: RISING

Dependencies: none
Supported: FW6.00 and later
Related: SCOT, SCOG, AUO1, DACO; 'Arm Digital Scope' Request Packet

Description: SCOE sets the trigger edge polarity of the internal digital oscilloscope. (Note: The internal scope uses the selected AUX_OUT1 signal as the trigger source. The scope captures 2048 samples of the signal selected by DACO. Also, the scope must be armed before a trigger can occur. See 'Arm Digital Scope' Request Packet.)

Errors: An INVALID PARAMETER error will be returned if the parameter is not one of the values specified.

Example:
```
SCOE=RISING;    // Trigger the internal scope on the rising edge of
                // of the trigger (selected AUX_OUT1 signal)
```

5.1.54 SCOG - Set Digital Scope Gain

Command: SCOG
Parameter: [1|4|16]
Units: none
Default: 1

Dependencies: none
Supported: FW6.00 and later
Related: SCOE, SCOT, AUO1, DACO; 'Arm Digital Scope' Request Packet

Description: SCOG sets the gain of the internal digital oscilloscope. (SCOG does not affect the DAC output.) The signal is not clamped for gains of 4 or 16; it will roll over, so care must be taken to interpret the results accordingly. (Note: The internal scope uses the selected AUX_OUT1 signal as the trigger source. The scope captures 2048 samples of the signal selected by DACO. Also, the scope must be armed before a trigger can occur. See 'Arm Digital Scope' Request Packet.)

Errors: An INVALID PARAMETER error will be returned if the parameter is not one of the values specified.

Example: SCOG=4; // Set the scope gain to 4x

5.1.55 SCOT - Set Digital Scope Trigger Position

Command: SCOT
Parameter: [87|50|12|-25]
Units: %
Default: 87%

Dependencies: none
Supported: FW6.00 and later
Related: SCOE, SCOG, AUO1, DACO; 'Arm Digital Scope' Request Packet

Description: SCOT sets the trigger position of the internal digital oscilloscope. 12%, 50% or 87% means that the trigger will be positioned at the 12%, 50% or 87% point of the digital scope data (i.e. the first 246, 1024 or 1782 samples are prior to the trigger and the rest are after the trigger.) -25% means that the trigger occurred 512 samples prior to the start of the scope data. (Note: The internal scope uses the selected AUX_OUT1 signal as the trigger source. The scope captures 2048 samples of the signal selected by DACO. Also, the scope must be armed before a trigger can occur. See 'Arm Digital Scope' Request Packet.)

Errors: An INVALID PARAMETER error will be returned if the parameter is not one of the values specified.

Example: SCOT=12; // Set the trigger position to 12%

5.1.56 SOFF - Set Spectrum Offset

Command: SOFF
Parameter: [OF{F}|{+|-}####.###]
Units: channels
Default: OFF
Range: Depends on number of channels specified by MCAC:
256 ch: -256 to +255.992 ch; 1/128 ch precision
512 ch: -512 to +511.984 ch; 1/64 ch precision
1024 ch: -1024 to +1023.969 ch; 1/32 ch precision
2048 ch: -2048 to +2047.937 ch; 1/16 ch precision
4096 ch: -4096 to +4095.875 ch; 1/8 ch precision
8192 ch: -8192 to +8191.75 ch; 1/4 ch precision

Dependencies: MCAC must be sent prior to SOFF
Supported: FW6.00 and later
Related:

Description: SOFF can be used to specify the spectrum offset, if desired. Applying a positive offset will shift the spectrum up (i.e. higher in energy), while a negative offset shifts the spectrum down.

Errors: An INVALID PARAMETER error will be returned if the parameter is not within the range specified.

Example: SOFF=10.5; // Shift the spectrum up by 10.5 channels

5.1.57 TECS - Turn On/Off the PC5 Thermoelectric Cooler (TEC) Supply and Set the Temperature

Command: TECS
Parameter: [###|OF{F}]
Units: K
Default: OFF
Range: 0-299K; ~0.1K precision

Dependencies: none
Supported: FW6.00 and later; not supported on DP5G

Description: TECS turns the PC5 TEC supply on or off, and sets the TEC temperature setpoint.

Errors: An INVALID PARAMETER error will be returned if the specified temperature is outside the range listed. A PC5 NOT PRESENT error will be returned if a PC5 is not detected.

Example: TECS=220K; // Turn on the TEC supply and cool to 220K
// or as cool as possible

5.1.58 TFLA - Select Flat Top Width

Command: TFLA
Parameter: [##.###]
Units: US (microseconds)
Range: The range is determined by the selected peaking time; see the table under the TPEA command.
Default: 0uS

Dependencies: TPEA must precede TFLA
Supported: FW6.00 and later
Related: TPEA

Description: TFLA selects the flat top width of the trapezoidal shaper. A flat top of 0uS will result in a triangular shape, not trapezoidal. The commanded peaking time (TPEA) determines the precision with which the flat top width can be commanded; the DP5 will round the commanded setting to the next lowest one.

Errors: An INVALID PARAMETER error will be returned if the commanded flat top is outside the range given in the TPEA table.

Example: TFLA=10; // This selects a flat top width of 10.00uS.

5.1.59 THFA - Select Threshold for Fast Channel

Command: THFA
Parameter: [###.##] {[###] in firmware prior to FW6.05}
Units: unitless
Range: 0-255.93 [256 equates to 50% of full-scale]
Precision: 1 part in 4096
Default: 0

Dependencies: none
Supported: FW6.00 and later
Related: PURE

Description: THFA sets the threshold for the fast channel. [Generally, the ideal fast threshold is determined empirically, or by using the 'Autoset Fast Threshold' request packet (which only works in the absence of a source.) This is discussed in detail elsewhere.]

The units or scale factor is somewhat problematic, because the fast channel doesn't have a digital fine gain control like the shaped channel does, so only changes in coarse (analog) gain affect the fast channel. The 'MCAS' (MCA Source) command can be used to route the fast channel into the MCA, so that a spectrum of the fast channel can be inspected.

The parameter is a multiple of 1/16. It ranges from 0/16 to 4095/16. This awkward-seeming arrangement was done to increase the precision of the setting, while maintaining backward compatibility to custom software, configuration files, etc.

Errors: An INVALID PARAMETER error will be returned if the commanded threshold is outside the range given.

Example: THFA=10; // This sets the fast threshold to 10.

5.1.60 THSL - Select Threshold for Shaped Channel

Command: THSL
Parameter: [##.###]
Units: % (percent of full-scale)
Range: 0-24.9% (with approx. 0.1% precision; approx. 0.01% precision for FW6.06 and later)
Default: 0

Dependencies: none
Supported: FW6.00 and later
Related: THFA, TLLD

Description: THSL sets the noise threshold for the peak detect unit. Events below this threshold will not be recorded in the spectrum; this acts as a low-level discriminator (LLD). [Use the 'TLLD' command in addition to 'THSL' if it is desirable to have the LLD reject more than just noise.]

Errors: An INVALID PARAMETER error will be returned if the commanded threshold is outside the range given.

Example: THSL=4.5; // This sets the shaped threshold to 4.5% of full scale

5.1.61 TLLD - Select Threshold for Low-Level Discriminator (LLD)

Command: TLLD
Parameter: [####|OF{F}]
Units: channels
Range: 256 ch: 0 - 255
512 ch: 0 – 511
1024 ch: 0 - 1023
2048 ch: 0 - 2047
4096 ch: 0 - 4095
8192 ch: 0 – 8191

Default: OFF

Dependencies: none
Supported: FW6.01 and later
Related: THSL

Description: TLLD sets the low-level discriminator (LLD) for the MCA; only events above this channel will be recorded in the spectrum. If the LLD is enabled (i.e. not OFF), then events must be above both the LLD threshold and the shaped (slow) threshold (see THSL) to be recorded.

Errors: An INVALID PARAMETER error will be returned if the parameter is outside the range 0- 8191; it is not error-checked with regard to the ranges listed.

Example: TLLD=110; // Events in channels 110 or lower are rejected

5.1.62 TPEA - Set Peaking Time

Command: TPEA
Parameter: [###.###]
Units: US (microseconds)
Range: 0.8-102.4uS (CLK=20MHz), 0.2-25.6uS (CLK=80MHz), 0.2-102.4uS (CLK=AUTO)
Default: None – this command is required.

Dependencies: CLCK (if used) must precede TPEA
Supported: FW6.00 and later
Related: TFLA

Description: TPEA selects the peaking time for the slow (shaped) channel. If CLCK=AUTO (automatic CLK selection) is commanded prior to TPEA, then TPEA will set the FPGA clock to 80MHz for peaking times of less than 5uS, and 20MHz for peaking times of 5uS or greater. The precision with which the peaking time can be programmed is listed below. [The selected peaking time also determines the range and precision of available flat top widths, so those are listed as well.] The DP5 will round the commanded peaking time to the next lowest one, so an error will only be returned if the commanded peaking time is outside the range listed above.

Errors: An INVALID PARAMETER error will be returned if the commanded peaking time is outside the range given.

Example:
```
TPEA=10;    // This selects a peaking time of 10.00uS. If
            // CLCK=AUTO was commanded prior to this, then
            // the FPGA clock will be set to 20MHz
```

FPGA Clock=20MHz				FPGA Clock=80MHz			
Peaking Time	**Precision**	**Flat Top**	**Precision**	**Peaking Time**	**Precision**	**Flat Top**	**Precision**
0.800 - 6.40uS	100nS	0.05 – 3.15uS	50nS	0.200 - 1.600uS	25nS	12.5 – 787nS	12.5nS
6.60 - 12.80uS	200nS	0.1 – 6.30uS	100nS	1.650 - 3.200uS	50nS	25 – 1575nS	25nS
13.20 - 25.60uS	400nS	0.2 – 12.60uS	200nS	3.30 - 6.40uS	100nS	50 – 3150nS	50nS
26.40 - 51.20uS	800nS	0.4 – 25.2uS	400nS	6.60 - 12.80uS	200nS	0.1 – 6.30uS	100nS
52.80 - 102.4uS	1600nS	0.8 – 50.4uS	800nS	13.20 - 25.60uS	400nS	0.2 – 12.6uS	200nS

5.1.63 TPFA - Select Peaking Time for Fast Channel

Command: TPFA
Parameter: [50100|200|400|1600]
Units: NS (nanoseconds)
Range: 50, 100 or 400nS for 80MHz FPGA clock; 200, 400 or 1600 for 20MHz FPGA clock.
Default: 100 for 80MHz clock; 400 for 20MHz clock

Dependencies: TPEA must precede TPFA
Supported: FW6.00 and later (50 & 200 settings require FW6.01 or later)
Related: THFA

Description: TPFA selects the peaking time for the fast channel. There are three settings available for each FPGA clock rate; only 400nS is available at both FPGA clock rates.

Errors: An INVALID PARAMETER error will be returned if the parameter violates the conditions listed under *Range*.

Example:
```
TPFA=400;   // This selects a fast channel peaking time of 400nS,
            // regardless of the FPGA clock.
```

5.1.64 TPMO - Select Test Pulser Mode

Command: TPMO
Parameter: [OF{F}|+S{NG}|+D{BL}| -S{NG}|-D{BL}]
Units: none
Default: OFF

Dependencies: none
Supported: FW6.00 and later
Related: DACF

Description: TPMO selects whether the Test Pulser is enabled, and if so, what type of pulses it produces. The test pulser uses the DAC output to simulate the output from a reset-style preamplifier. It produces a step in the DAC output at a rate of 2kHz. This signal is intended to be connected directly to the DP5 signal input for test and diagnostic purposes.

If +SNG or –SNG is specified, the DAC step is 1/4 of the value specified by the DACF command. (i.e. for DACF=200MV, "+SNG" produces +50mV steps and "-SNG" produces -50mV steps.) If +DBL or –DBL is specified, two step sizes are produced alternately; one is ¼ the value specified DACF, the other is 1/16 the value. (i.e. for DACF=200MV, "+DBL" produces +50mV and +12.5mV steps alternately; "-DBL" produces -50mV and -12.5mV steps alternately.)

For really large steps, a negative DACF can be used. This is treated as an unsigned value; to calculate the resulting step size from a negative DACF setting, use 1V+(DACF parameter). For example, for DACF=-200MV, +SNG produces ¼ (1V + -200mV) = 200mV steps.

Detector Reset Lockout (RESL) should be disabled or set to a short interval when the test pulser is enabled, since a detector reset is simulated when the DAC output wraps around. (It produces a large pulse of the polarity opposite the desired pulses, which looks like a detector reset.)

Note that if the Test Pulser is enabled, the internal digital scope will still capture the signal specified by DACO, even though the DAC will output test pulses.

Errors: An INVALID PARAMETER error will be returned if the parameter is not one of the values specified.

Example:
```
TPMO=+SNG;          // Enable the test pulser to produce single
                    // positive-going steps
```

5.1.65 VOLU – Turn the PX5 speaker ON/OFF

Command: VOLU
Parameter: [ON|OF{F}]
Range: ON or OF{F}
Default: OFF

Dependencies: none
Supported: FW6.01 and later; PX5 only, not supported by DP5 or DP5G, which don't have speakers

Description: This turns the PX5 speaker on or off. This simulates a Geiger counter, and produces a click for each event recorded by the fast channel.

Errors: An INVALID PARAMETER error will be returned if the parameter is not ON or OF{F}.

Example: VOLU=ON; // Enable the PX5 speaker